SEMI-CONTINUOUS HIDDEN MARKOV MODELS FOR AUTOMATIC SPEAKER VERIFICATION

Mark Eric Forsyth

A thesis submitted for the degree of Doctor of Philosophy

1995

Acknowledgements

Thanks are due to many whose help I am very grateful to have received over the last few years. To begin with there is my source of funds, the Commonwealth Scholarship Commission. There is no doubt that without the assistance of such a scholarship it would not have been possible to study at Edinburgh University.

I wish to thank my first supervisor, Professor Mervyn Jack for his supervision, for ensuring the opportunity was always available for me to travel to conferences and learn from other researchers around the world, and especially for giving me enough freedom to pursue my ideas and the facilities to get the research done. Along with Professor Jack, my colleagues Fergus McInnes, Paul Taylor, and Alan Wrench have helped in reviewing the manuscript of this thesis, for which I am extremely grateful. All remaining errors and omissions are, of course, solely my responsibility. Thanks to British Telecom for allowing me to use their speech database, and to the computing officer, Bob Anstruther for his assistance.

The speech research community at 80 South Bridge changes constantly, and I believe that is one of the benefits of starting a career in research there. Although excellent people leave, other are constantly arriving and a PhD student can learn from them all. Over the years I have had advice and assistance from many people here. Paul and Alan showed me the ropes and taught me to program a little better. Fergus has always been the one I, and so many other people, go to for the *right* answer. His ability to understand what I *really* meant to say always amazes me. Paul Bagshaw was about a year ahead of me in the PhD process, and I shamelessly borrowed all his hard-won knowledge of Latex, Splus, and encapsulated post-script. I'm very grateful for that, and for his friendship and wisdom as a fellow post-grad.

Cheers also to the gang from the Edinburgh University Volleyball Club, for the sport, the perspective and the laughs. People say a PhD can be agony, but I can't agree. Hard work it undoubtedly is, but I've had a great time -fun, interesting and challenging. Part of that is due to being in Edinburgh, which is a wonderful city, but mostly it is due to my friends. Thank you all. Finally, I'd like to thank my family, especially my mother, who made New Zealand seem not so far away.

Dedication

This thesis is dedicated to my family.

"To begin with," he said heavily, "you've got to understand that a seagull is an unlimited idea of freedom, an image of the Great Gull, and your whole body, from wingtip to wingtip, is nothing more than your thought itself. . . .Let's begin with Level Flight . . ."

Jonathan Livingston Seagull (Bach, 1970)

Contents

List of figures

List of tables

Glossary

The following list defines all the important terms in this document. They are all defined in the text on their first occurrence.

ACW Adaptive Component Weighting. A cepstral weighting scheme.

ASR Automatic Speech Recognition.

ASI Automatic Speaker Identification.

ASV Automatic Speaker Verification.

CHMM Continuous Density Hidden Markov Model.

Casual impostor The impostor is using their natural voice and are not trying to mimic the client speaker.

Client speakers speakers who are enrolled on, and modelled by, the ASV system.

Delta, Δ first order differential of a feature. In this document the Δ is calculated over a window of 5 frames.

DHMM Discrete Hidden Markov Model.

DOP Discriminating Observation Probabilities.

DTW Dynamic Time Warping.

EBI bias Eliminating best impostor bias. A form of experimental bias in assessment of cohort normalisation schemes, which is due to eliminating the most successful impostors from the test database.

EER Equal Error Rate. The threshold is set *a posteriori* to ensure FR rate equals the FA rate. The threshold can be speaker specific or speaker independent.

EII bias Experimentally invalid impostor bias. A form of experimental bias in assessment of cohort normalisation schemes, which is due to using impostors who are explicitly modelled.

FA False Acceptance. Test utterances from an impostor which are accepted by the ASV system are known as FA errors (also called Type II errors).

Feature A frame-based feature, usually a vector, extracted from speech e.g. LPC cepstra.

FFT Fast Fourier Transform.

FR False Rejection. Test utterances from the client speaker which are rejected by the ASV system are classified as FR errors (also called Type I errors).

GPD Generalised Probabilistic Descent. An optimisation scheme for discriminative training of HMMs.

GMM Gaussian Mixture Model. Single state CHMM.

HMM Hidden Markov Model.

Impostor speaker making a false identity claim. Usually a casual impostor, unless otherwise stated.

LDA Linear Discriminant Analysis.

LPC Linear Predictive Coding.

LSP Line Spectral Pair.

MCE Minimum Classification Error. A discriminative training scheme for HMMs.

MFCC Mel Frequency Cepstral Coefficients.

MLE Maximum Likelihood Estimation. The standard optimisation criterion for training HMMs.

MMIE Maximum Mutual Information Estimation. A discriminative training scheme for HMMs.

NN Neural Network.

PDF Probability Density Function.

RASTA RelAtive SpecTrAl. Signal processing technique to make features robust to channel variation.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic. Plot of correct acceptance versus false acceptance as the verification decision threshold is varied.

SCHMM Semi-Continuous Hidden Markov Model, also known as tied-mixture continuous HMM.

SI EER Speaker Independent Equal Error Rate. The decision threshold is the same for each (client) speaker.

SS EER Speaker Specific Equal Error Rate. The decision threshold is possibly different for each (client) speaker.

TDNN Time Delay Neural Network.

Threshold Probability used for the verification decision. Probabilities greater than or equal to the threshold will be accepted, probabilities below the threshold will be rejected.

Type I errors False Rejection (FR) errors.

Type II errors False Acceptance (FA) errors.

TD Text-Dependent. The text of the test utterance is constrained in some way by the ASV system. All text-prompted systems are text-dependent if the constraints on the speech, or knowledge of the text of the speech are used by the system. A text-dependent system cannot be used on a text-dependent task.

TDM Targeted Distance Measure. Performance measure for ASV systems based on the distance between impostor and client probability distributions.

TI Text-Independent. The text of the test utterance is not constrained by the ASV system in any respect, except perhaps for the length of the utterance. If the ASV system determines the text of the test utterance automatically (by speech recognition) the ASV system is still text-independent, provided no constraint is placed on the text of the test utterance. Any text-independent system can be used on any text-dependent task.

VQ Vector Quantisation. A discrete number of labels are used to quantise a vector space. Any vector in that space is represented by the label that it is closest to, according to some distance measure.

YOHO An ASV telephone speech database.

ZFR Zero False Rejection. The ZFR rate is the minimum false acceptance rate when the threshold is chosen so that the false rejection rate is zero.

ZFA Zero False Acceptance. The ZFA rate is the minimum false rejection rate when the threshold is chosen so that the false acceptance rate is zero.

Chapter 1

Introduction

In a world rich in opportunity, time is the key constraint to our activities and time and convenience are highly valued. We cannot increase the number of hours in a day so we leverage our technological capabilities to improve the quality of the day.

The yellow pages business directories encapsulated this philosophy in the slogan *let your fingers do the walking*. This encourages people to use telephone technology to save themselves time in locating business services. Instead of physically looking for a product in several locations, the search can be conducted remotely and the only travel necessary is for the final purchase. The next step, of course, is to perform the actual transaction remotely as well. So compelling are the forces driving this innovation that telephone based transactions have achieved widespread popularity and acceptance before it was technologically possible to make transactions properly secure. Lost or stolen credit card numbers can be used to purchase a wide range of goods with reasonable anonymity. Telephone-based credit-card fraud costs millions of pounds every year.

An effective way to reduce credit-card fraud at the point of sale is to etch the account holder's photograph onto credit cards, but this approach is of no use over the telephone. Personal identity numbers and passwords can be used to increase security in telephone transactions but they can be forgotten, and if recorded they can be stolen.

Biometrics are measures based on physical characteristics of a person and as such cannot be lost, stolen, borrowed or forgotten. This makes them very convenient for security purposes. Biometrics such as fingerprints, iris patterns, and hand shapes can be employed to restrict physical access, but cannot be easily used over the telephone. The natural, and most convenient biometric for telephone transactions is the speaker specific content of the human voice.

A person's speech contains many different types of information. The primary information

is lexical, but information about the person's background (dialect, education, native language), their emotional and physical state (stress, tiredness, illness) and the physical structure of their vocal apparatus are also encoded in the speech signal. Many of these factors are speaker-specific and can be used to discriminate between speakers. While it is not known whether a person's speech characteristics are in fact unique, the amount of inter-speaker variation relative to intra-speaker variation is sufficient to discriminate between speakers with a useful degree of reliability.

Automatic speaker recognition involves identifying people from their voices completely automatically. For telephone transactions such as banking or shopping, the requirement is to verify that a caller is who they claim to be. This task is known as automatic speaker verification (ASV). The immediate goal is to provide a level of security which, when added to security measures already in place, will make it possible to perform all shopping and banking transaction over the telephone, rather than in person. ASV for telephone applications is the focus of the research presented here.

There are four main chapters in this thesis. Chapter 2 introduces the field of automatic speaker recognition. The motivation for researching the field that comes from the actual and potential applications is discussed.

Various measures of performance are available for use in evaluating ASV systems, each of which measures a different aspect of performance. These are discussed in Section 2.2.

This work was conducted with a telephone banking task in mind, but even with that constraint there are a multitude of areas for variation in the task definition. These areas are described in Section 2.3. Many of these variations in task definition can lead to significant variation in error rates. Which tasks are easier and which are harder is discussed, with reference to the literature.

Unfortunately a standard telephone-based ASV task has not yet emerged in the literature, with the result that almost every system performs a different task and so it is almost impossible to compare results between papers. The only valid comparison of algorithms is a series of comparative experiments using a single ASV system on a single database. Even then, if the algorithm is applied to another task the assumption has to be made that the relative performance of the techniques will hold if the task is changed.

Speech modelling is of critical importance to ASV and the strengths and weaknesses of the main model architectures are discussed in Section 2.4.2. Chapter 2 concludes with some

discussion of ASV systems from the recent literature.

Chapter 3 describes the hidden Markov model automatic speaker authentication system (HASAS) which was developed and implemented by the author. A detailed task definition is given along with a description of the database used to evaluate HASAS. The task definition and database specification are then combined to produce a system specification which includes the details of the feature sets and modelling used.

A notable aspect of the system design is the use of a common segmentation across all feature sets throughout training and verification. This was done to ensure a clearer assessment of the relative merits of different feature sets.

Chapter 4 describes a series of experiments performed using HASAS which have an underlying goal of extracting as much information as possible out of the modelling stage of the system. Standard feature sets and a simple linearly thresholded decision logic are used. A baseline system using LPC cepstra-based models is established which has good performance. The use of digit weights and multiple codebooks cut the error rate almost in half compared with that obtained using the baseline system.

The focus on the modelling stage of the ASV system is continued in Chapter 5 with the development of a new discriminating model architecture known as discriminating observation probability (DOP) HMMs which were spawned from the idea of using a common state segmentation.

The DOP architecture involves the construction of a discriminating model from two standard models, without the need for discriminative training. It is a very flexible architecture with potential application to other binary classification problems. The application of DOP models to ASV is described in Section 5.3, and evaluated in the remainder of Chapter 5.

The relationship of DOP modelling to the so-called speaker normalisation techniques currently popular in the literature is investigated in Section 5.8 and its superiority is shown experimentally.

Three key areas of the DOP architecture are the choice of impostor model, the choice of segmentation model and the choice of discriminating function. The choice of impostor model has recently been studied in the literature in the context of log-likelihood normalisation. This topic is discussed at length in Section 2.4.4. The choice of segmentation model is discussed in Section 5.6 and two choices are compared experimentally. The choice of discriminating function

is discussed in Section 5.9 and some areas for potentially fruitful research are proposed.

The final chapter summarises the findings of Chapters 4 and 5.

Chapter 2

Automatic Speaker Verification

2.1 Task Definition

Speaker recognition is the task of recognising people by their voices. For some applications, notably forensics, human experts are often employed to perform speaker recognition. For over 30 years research has been conducted into ways to perform this task automatically using computers.

Many excellent reviews of automatic speaker recognition are available (Rosenberg *et al.*, 1992; Furui, 1994; Naik, 1994; Bimbot *et al.*, 1994; Doddington, 1985; O'Shaughnessy, 1986; Foil & Johnson, 1983; Rosenberg, 1976).

Automatic speaker recognition can be classified into two closely related tasks, automatic speaker verification (ASV) and automatic speaker identification (ASI). ASV is concerned with the classification of unknown bidders into two classes, client speaker or impostor. There is an initial enrolment procedure in which a *client model* is constructed from speech data supplied by the client. The verification process consists of an utterance being supplied along with an identity claim. The verifier either accepts or rejects the claim. ASI, on the other hand, does not involve an identity claim and requires a *most likely* decision from a list of N possible speakers. For a given ASI system the error rate depends on the number of people in the identification set (N). As N tends towards infinity the error rate will tend to towards 100%[1]. A further classification of ASI can be made into closed-set and open-set tasks. The open-set ASI task allows for the possibility that the utterance did not come from any of the speakers in the identification set. This is essentially closed-set ASI followed by ASV.

[1] An ASV system, on the other hand, should have the same error rate, regardless of how many speakers are enrolled on the system.

Obviously an ASV system which produces a verification score rather than a binary decision can be used for ASI by simply ranking the scores from the models of each of the speakers in the identification set. If this approach is used then the fields of ASI and ASV are effectively the same. The two fields only really diverge when an ASI system takes advantage of knowledge of the identification set, perhaps to train models which explicitly discriminate against other speakers in the identification set.

ASI is a popular task in the research community, but most applications of automatic speaker recognition are for ASV. Doddington (Doddington, 1985) goes as far as to say

> *It is difficult for me to visualise a real operational application of speaker identification yet the identification task formulation remains popular in laboratory evaluations.*

Since 1985 some real operational applications of ASI have emerged. One realistic ASI application which has been proposed is automatic segmentation of multi-speaker speech (Wilcox *et al.*, 1994) (Yu & Gish, 1993), which would be required for automatic transcription of parliamentary or court proceedings. Another possible application is automatic classification of voice mail by speaker, although classification according to the calling number might be a cheaper and less sophisticated way to achieve the same goal.

(Bennani & Gallinari, 1991) propose the use of ASI to select the best set of semi-speaker-independent speech recognition models, to improve the performance of a speech recogniser. This avoids the need to train or adapt speaker dependent models for a complete range of sub-word units. The ASI system used would be text-independent, so it could be trained on the first few minutes of a new speaker's speech. Forensics is an area where open-set ASI is used (Federico & Paoloni, 1993) in an assessment which follows the same principle as the traditional identity parade.

2.2 Measuring Performance

There are two types of correct classification, the acceptance of client speakers, and the rejection of impostors. There are two corresponding types of errors, namely the rejection of genuine speakers, called *false rejection (FR)* or *TYPE I* errors, and the acceptance of impostors, called *false acceptance (FA)* or *TYPE II* errors.

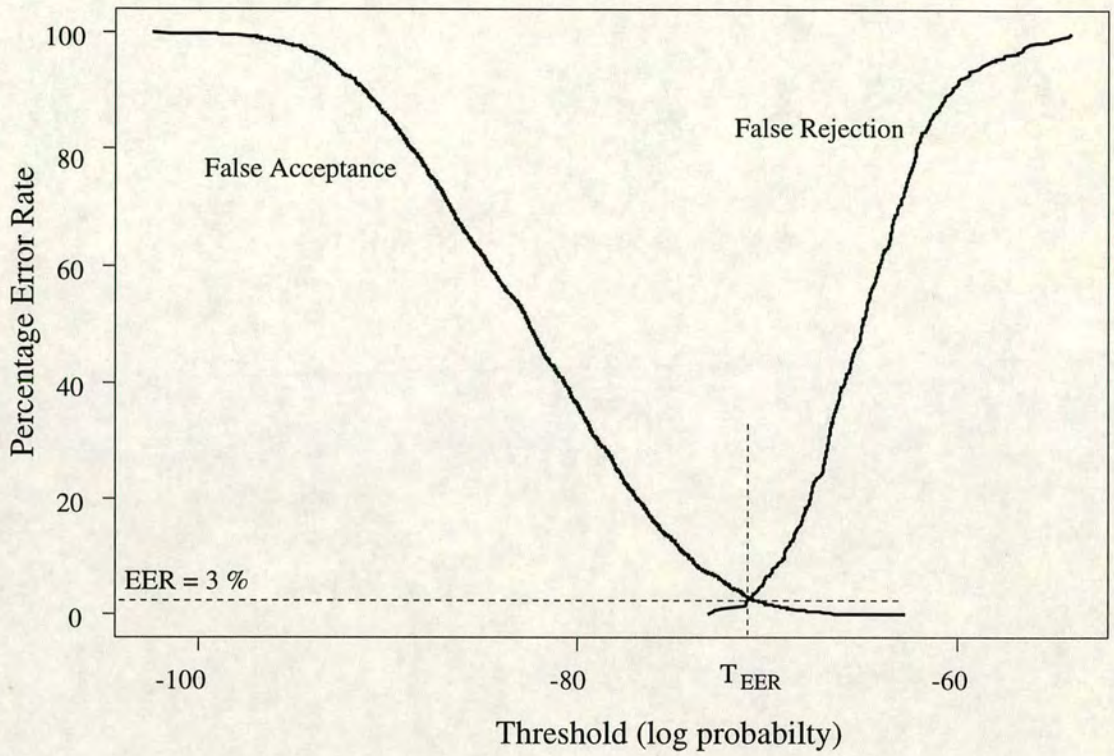

Figure 2.1: Typical plot of FR rate and FA rate against choice of decision threshold. The EER, ZFR, and ZFA can be determined from this plot.

The verification decision is made by applying some form of threshold to a *verification score*. The verification score is some measure of the match between the client model and the test utterance. The thresholding is usually explicit but it can be implicit, as in some neural net systems.

2.2.1 Error Rates

Figure 2.1 is a typical plot of FA rate and FR rate against the choice of decision threshold. Notice that there is a trade-off between FR and FA. Error rates for any given threshold can be determined from this plot.

Several commonly used error measures can be determined from this plot, and they are described in the following two sections.

Zero False Rejection and Zero False Acceptance

The Zero False Rejection (ZFR) Rate is the FA rate when no genuine speakers are rejected and the Zero False Acceptance (ZFA) Rate is the FR rate when no impostors are accepted. These measures are critically dependent on the worst client speaker score and the best impostor score, respectively. The ZFR and ZFA measures cannot be used as the sole basis for selecting one algorithm over another, since slight changes in the data could easily reverse the rankings of the algorithms, as was demonstrated in (Forsyth & Jack, 1994).

Equal Error Rate

The most common performance measure referred to in the literature is the equal error rate. This involves applying an *a posteriori* threshold T_{EER} which makes the percentage of FA and FR errors equal. It is defined in Figure 2.1 by the point where the FR and FA curves cross.

It is important to make a distinction between whether T_{EER} is speaker-specific (SS) or speaker-independent (SI). If the EER is calculated for each speaker separately then T_{EER} is speaker specific. If the same T_{EER} is used for all speakers then the EER is speaker independent. Speaker specific EERs (SS EER) are quoted as an average over all client speakers and this average tends to be considerably lower than speaker independent EERs (SI EER). Both SI and SS EERs are used in the experiments of Chapters 4 and 5. T_{EER} can also be text-specific in applications where the text is known.

The use of an EER implies an optimum choice of threshold, which is not possible in a real application since the threshold would have to be determined *a priori*. This is particularly true for SS EERs since the threshold for each speaker must be estimated at enrolment time, and it has not yet been determined how this can be achieved reliably, given the limited amount of client speech data that is available at enrolment. The use of an *a posteriori* threshold means that the EER provides an upper bound on performance and does not indicate how robust the system is to an imperfect choice of threshold.

2.2.2 Receiver Operating Characteristic (ROC)

Another measure of performance often used in ASV is the Receiver Operating Characteristic (ROC).

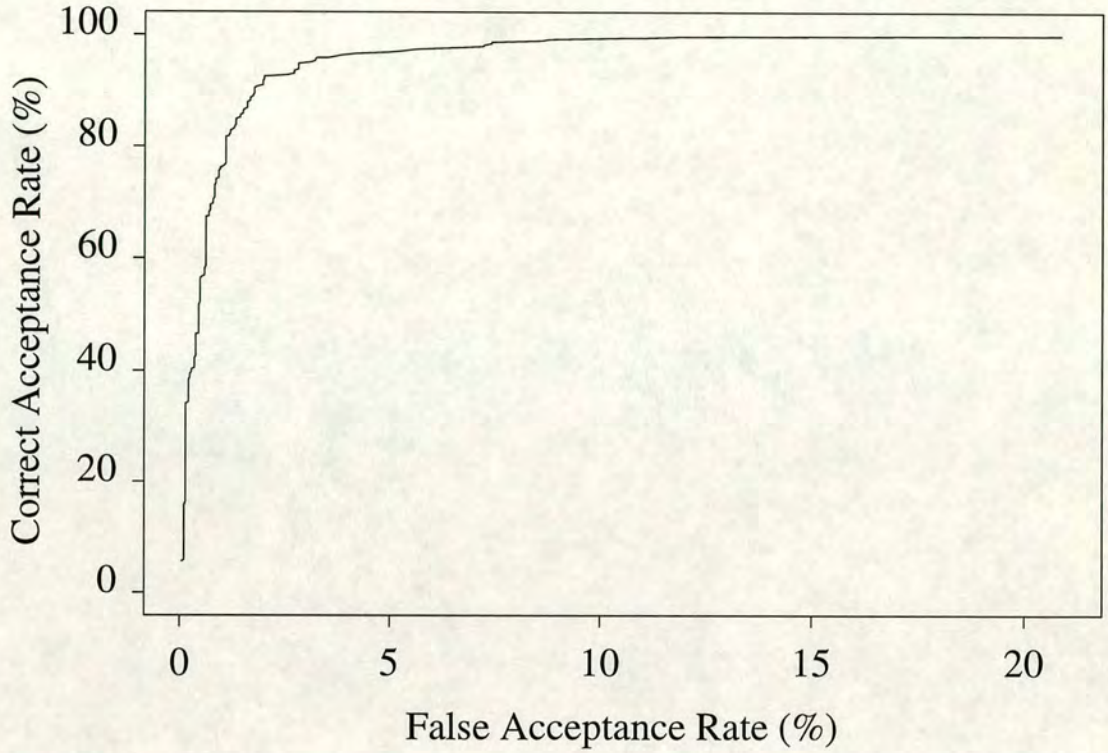

Figure 2.2: Typical receiver operating characteristic. The y-axis is the correct acceptance percentage (100-FR), and the x-axis in the percentage false acceptance (FA). The curve indicates the balance between the two error types as the threshold is varied.

This is a measure of correct acceptance rate against false acceptance rate (Furui, 1994). An example of an ROC curve is shown in Figure 2.2. The ROC curve gives a good representation of the trade-off between FA and FR errors and can be used to select an appropriate operating point for a particular application. In order to quantify the information contained in the ROC, so that different systems can be compared, a parametric representation of the ROC was recently proposed (Oglesby, 1994).

2.2.3 Distance Measures

Although EER is an important performance measure, it is of little use when error rates are very low or zero, because algorithms cannot be compared with sufficient statistical confidence in such cases. For this reason it is also useful to have a measure of how well a system separates the probability distributions for the client speakers and the impostors. Measures based on the distances between client and impostor scores perform this role, although they are not commonly used in the ASV literature. This section describes two distance measures, the Mahalanobis

distance and a new *targeted* distance measure (TDM) which is used to evaluate performance in some of the experimental work which follows. Such measures give an indication of the robustness of the system to an imperfect choice of threshold, and are especially useful when the number of errors is small or zero.

Mahalanobis Distance

The Mahalanobis distance (MD) is a parametric measure of the distance between two statistical populations (Mahalanobis, 1936), which assumes that the two populations have normal (Gauss-Laplacian) distributions. Consider that the two populations of log probabilities from impostor ($i = 1$) and client ($i = 2$) scores are respectively represented by the sets,

$$x_i = \{x_{i,k} | k = 1, 2, \ldots, N_i\} \quad i = 1, 2 \tag{2.1}$$

An experimental evaluation shows that these score populations are normal distributions with a Lilliefors' probability (Lilliefors, 1967) of approximately one. Note, however that the score populations deviate most from normal distributions above the 90^{th}-percentile for the impostor scores and below the 10^{th}-percentile for the client speaker scores, and these are the scores which are likely to be involved in classification errors. Assuming the client and impostor score populations are univariate normal distributions, the Mahalanobis distance between these two populations is given by,

$$D^2 = \frac{(\bar{x}_1 - \bar{x}_2)^2}{\sigma_{12}} \tag{2.2}$$

where,

$$\bar{x}_i = \frac{1}{N_i} \sum_{k=1}^{N_i} x_{i,k} \quad i = 1, 2$$

$$\sigma_{12} = \frac{1}{N_1 + N_2 - 2} \sum_{i=1}^{2} \sum_{k=1}^{N_i} (x_{i,k} - \bar{x}_i)^2$$

The MD gives a measure of the separation between client speaker scores and impostor scores. Unfortunately, as was shown in (Forsyth *et al.*, 1994), this is not an ideal measure for the purpose of quantifying speaker discriminating performance of an ASV system. This is because the primary goal of a new algorithm is to reduce verification errors and most impostors are

never mistaken for genuine speakers and most genuine speakers are not usually falsely rejected. Thus, the scores which most need to be improved are those near the verification threshold. The Mahalanobis distance assigns equal importance to all scores. A distance measure which targets the most important scores is required.

Targeted Distance Measure

A figure of merit called the *targeted distance measure* (TDM) has recently been proposed and evaluated (Forsyth *et al.*, 1994). TDM targets the most important scores, namely the highest third of the impostor scores and the lowest third of the genuine speaker scores. It is calculated by the addition of two distance measures — TDM_I for the impostor scores and TDM_C for the client speaker scores.

$$TDM = TDM_i + TDM_c \tag{2.3}$$

where,

$$TDM_I = 100. \left[\frac{1}{|\bar{x}_1 - \bar{x}_2|} \cdot \frac{3}{N_1} \sum_{k=\lceil 2N_1/3 \rceil}^{N_1} (T_{EER} - \hat{x}_{1,k}) \right]$$

$$TDM_C = 100. \left[\frac{1}{|\bar{x}_1 - \bar{x}_2|} \cdot \frac{3}{N_2} \sum_{k=1}^{\lfloor N_2/3 \rfloor} (\hat{x}_{2,k} - T_{EER}) \right]$$

$$\hat{x}_{i,k} = k^{th}\text{member of } x_i \text{ sorted in ascending order}$$

This calculation takes an average *signed* distance from T_{EER} and normalises it with respect to the distance between the means of the two distributions. Note the reversal of sign between the calculation of TDM_I and that of TDM_C, so that a higher number corresponds to better performance in both cases.

2.3 Classification of ASV Tasks

Within the definition of the general ASV task defined in Section 2.1 there is scope for a wide variety of tasks to suit a multitude of applications.

Factors which can vary include

- Client speakers. The number, sex, age , level of co-operation and dialect variation of the speakers who use the system.
- Training Data. The quantity, quality and content of training data can vary considerably depending on the application.
- Recording environment. Including microphone type and position , background noise
- Storage Requirements. Some applications impose significant restrictions on the amount of data that needs to be stored to model each client speaker (such as those where the speaker's data is stored on a magnetic card.)
- Transmission Channel. Many applications involve the speech signal being transmitted across a standard telephone channel, with the associated problems of limited bandwidth and channel and microphone variation. An even more difficult channel is that encountered when using cellular phones.

All of these factors influence the difficulty of the task or application to varying extents. The relative importance of these factors is difficult to quantify precisely. Most ASV systems reported in the literature are applied to tasks defined by specific combinations of these factors, but the combination is almost never the same, making comparison between systems extremely difficult. Refer to (Millar *et al.*, 1992; Oglesby, 1994) for a discussion of this topic. A standard task has recently been proposed (Campbell, 1995) based on the YOHO database (Godfrey *et al.*, 1994), but although the database is recorded using a telephone handset, there is no variation in the handset type and the data are not recorded over a telephone channel. This limits its appropriateness for ASV systems designed for telephone applications.

The following section describes some of the different aspects of an ASV task which can affect performance and must be taken into account when comparing systems.

2.3.1 Text-Dependent versus Text-Independent

ASV tasks are often divided in the literature into *text-dependent* and *text-independent*. Unfortunately there has been some variation in the definition and use of these terms.

The distinction between TD and TI is made in order to separate different tasks and therefore the most useful definition will be one based on tasks, or the way the system is used rather than the system itself.

TI tasks include forensics where the client is uncooperative, and *background* verification where the speaker is monitored in the course of a normal conversation and where the client may well be cooperative but is unlikely to be making any effort to speak clearly or consistently, or to place any constraint on their vocabulary.

TD tasks usually involve some form of pre-determined or prompted password, in order to obtain the required text. The client can generally be expected to be co-operative, and will also endeavour to speak clearly and consistently so that they will be correctly accepted by the system. Conscious mimicry is more likely in TD systems than TI systems.

Rosenberg (Rosenberg, 1976) suggests that text-independent systems must place no constraint on the content or length of the test utterance and all other systems are text-dependent. This is the definition one would assume from the terms TD and TI.

Full text-independence is a necessary requirement for many forensic applications where test speech may be collected under completely uncontrolled conditions.

It is difficult, however, for many automatic system to not place any constraint on the length of the test utterance, as many are based on long term averages which require a minimum duration to stabilise.

Other so called text-independent techniques rely on extracting certain specified speech events which can be extracted for analysis. The speech events are extracted automatically and no knowledge of the text is required, but a constraint has nevertheless been imposed that the text be general enough to contain the required speech events. This constraint is a very reasonable one for applications involving the monitoring of text of any significant length, so they can usefully be described as TI.

There are several *text-flexible* systems which are described as TI. Often they are based on sub-word models of phones or tri-phones, from which any text can be constructed. These could potentially be text-independent - if the phone models used are determined using automatic speech recognition (ASR). If knowledge of the text is used to select the correct models then the task is clearly TD.

The requirement for text-independence seems to apply only to the test data and not training

data. Certainly for a complete set of sub-word models to be trained a significant amount of phonetically rich data must be available for training. This implies either a specified training text or a very large amount of training data.

In many TI applications it is possible for the training to be TD. In forensics, for instance, a suspect will sometimes be available to supply training data in whatever form is required, and may be prompted for certain key words.

In the case of a bank wishing to monitor the identity of its client during a transaction or negotiation it is likely that the client could provide whatever training data was required. It is possibly only in the case of surveillance operations that the nature of the training data cannot be specified.

The following definitions are applied to the ASV systems reviewed in this chapter.

- Text-Independent. The text of the test utterance is not significantly constrained by the ASV except for the length of the utterance. If the ASV system determines the text of the test utterance automatically (by speech recognition) the ASV system is still text-independent, provided no constraint (other than a minimum length) is placed on the text of the test utterance. Any text-independent system can be used on any text-dependent task.

- Text-Dependent. All systems which are not text-independent are text-dependent. All text-prompted systems are text-dependent if the constraints on the speech, or knowledge of the text of the speech are used by the system. A text-dependent system cannot be used on a text-independent task.

All other factors being equal TI tasks are more difficult than TD tasks (Soong & Rosenberg, 1988) and error rates are correspondingly higher. The system described in this thesis is intended for TD applications.

2.3.2 Recording Conditions

There are several ways in which the recording conditions can vary from one database to another. Firstly there is the sampling rate, which must be at least twice the bandwidth of the speech signal.

Another area of variability is the background noise from such things as fans, traffic, speech and music. Speech databases have been recorded in many environments such as a car, an open-plan office, a private office and a sound booth. The *cleaner* the data the easier the verification

task becomes. The type of microphone used is also important, since each microphone will colour the spectrum of the speech signal in a unique way (Wang *et al.*, 1993). Close-talking microphones reduce background noise more than hand-held or screen mounted microphones, which in turn have better frequency response than telephone microphones.

Consistency in the recording environment is also important. Ideally the level and type of noise should be the same in the training data as in the test data. Mismatches in the training and testing conditions cause a decrease in performance (Openshaw *et al.*, 1993).

Telephone Speech

ASV has an advantage over most other forms of security for telephone application. Many other biometrics such as fingerprints, DNA, hand-size and iris patterns cannot be easily used. Tone dialled personal identification numbers (PIN) can be used but they are not very secure. For this reason the most appealing applications of ASV technology involve the telephone and so ASV systems are often designed to cope with a telephone channel.

There is ample evidence in the literature that ASV is considerably more difficult telephone speech is used (Irvine & Owens, 1993; Reynolds, 1994).

Telephone speech is often sampled at 8kHz because the telephone channel acts as a low-pass filter with a cut-off around 3.5kHz. It is important to filter telephone speech to make the pass-band consistent between recordings (Gish, 1990). Gish uses a band-pass filter with a pass band from 300Hz to 3300Hz. The telephone channel eliminates the higher frequencies in the speech signal - frequencies which have been shown to have important speaker discriminating information (Hayakawa & Itakura, 1994).

The spectral characteristics of the telephone channel vary from moment to moment and from call to call. Local calls, long-distance calls and cellular telephone calls all create widely varying channel characteristics. Apart from spectral distortion, the telephone channel can introduce line noise and various clicks and pops. The handset microphone also introduces spectral distortion (Wang *et al.*, 1993), as well as breathiness in the speaker and creaks from the handset itself. The use of speaker phones is a particularly challenging telephone application because of the feedback from the loudspeaker. Cellular phones are often used in very noisy environments, and speech detection features can make measuring background noise levels very difficult since they cut off the signal whenever there is no speech (Raman & Naik, 1994).

An important factor in recording a telephone-speech database is to record the data over several sessions and on different handsets. Not only does this capture realistic intra-speaker temporal variation, but it also increases intra-speaker channel variation and so prevents the ASV system becoming a *channel recogniser* rather than a speaker-recogniser.

2.3.3 Impostors

Bidders whose identity is not that of the client they claim to be are known as *impostors*. When evaluating an ASV system a set of impostor speakers is used to test the correct rejection and false acceptance levels of the system.

The make-up of the impostor set should be appropriate to the task and has significant bearing when two ASV systems are being compared. The following sections describe the important characteristics which make one impostor set different from another.

Sheep and Goats

It is well known that in any given database there will be variation in performance among the speakers.

Some speakers have no difficulty being correctly accepted by the system and are difficult for impostors to impersonate. These speakers are known as *sheep*, and it clear that in some sense their voices are distinctive and/or consistent[2], although it is difficult to isolate exact reasons for their success.

Other speakers have difficulty using the ASV system, are often falsely rejected, and are easier to impersonate. These speakers are known as *goats* and they must have highly variable voices and/or very common voice characteristics. So distinct is the performance difference between sheep and goats that a method of pre-screening clients at enrolment in order to pay special attention to goats has been proposed (Thompson & Mason, 1994).

Most of the errors in the evaluation of an ASV system are caused by the goats, and so the overall error rate is largely determined by the proportion of goats in the database[3]. This makes comparisons between databases very difficult. In particular a single goat can dominate

[2] Corresponding to large inter-speaker distance and/or small intra-speaker variation.

[3] Ideally the database is large enough that the proportion of goats in the test database matches that of the general population.

performance measures such as the ZFR rate and the ZFA rate, and so cross-database comparisons are not valid using these performance measures.

Applications of ASV such as telephone banking face potential impostor populations of thousands or even millions of people. It is very difficult to collect a database which is truly representative of such a population, but any database for evaluation of an ASV system, should be realistically variable and large enough to make the results statistically significant.

Dedicated and Casual Impostors

In practice most impostors can be assumed to be trying to deceive the system. If they have no knowledge of the client speaker they claim to be, however, there is very little they can do, and it can be assumed that they will speak in their usual voice. Such impostors are known as *casual impostors*.

If the impostor has knowledge of the client's voice and uses that knowledge in his or her bid then they can be considered a *dedicated impostor*. Mimics (both untrained and professionals), identical twins and family members have all been used as dedicated impostors in studies.

Most ASV systems reported in the literature use casual impostors. Whether this reflects a realistic test is debatable and would depend on the application. In the case of telephone shopping by credit card it is likely that most of the fraud occurring today would involve fraudsters with no knowledge of the card owners voice. If an ASV system were to be employed, it is likely that most impostors would be casual impostors. If an ASV system is being used in door-entry or computer log-in systems, however, it is more likely that the impostors will have knowledge of the client's voice, so such systems should at some point be tested against dedicated impostors.

Some attempts have been made to quantify the difference in performance between casual and dedicated impostors. Several experiments using mimics as impostors are reviewed in (Rosenberg, 1976). The mimics had varying success, with increases in false acceptance rate of 100-150%. The experiments were highly favourable to the impostor, since immediate feedback of an mimic's success was given during "practice" sessions and the mimics themselves were highly skilled professionals.

A study using identical twins and other sibling pairs found that they were only slightly more successful as impostors than average (Oglesby, 1994). Interestingly, in this study the dedicated impostors were significantly less successful than the best casual impostors.

Male and Female Impostors

Virtually all applications of ASV will require that both male and female speakers be enrolled as clients. It can be assumed that impostors could be either male or female. It is uncommon for a male impostor to be accepted as a female client and vice-versa. Soong reports between 9.4% and 26.4% of errors being cross-sex, depending on the configuration of the system (Soong & Rosenberg, 1988).

In the case of credit card fraud referred to previously, there is a possibility that if a first name or title (and not just initials) is printed on the card, then the sex of the client could be known. Casual impostors would then be likely to be of the same sex as the client, because they would tend not to attempt to defeat the system if they were not of the correct sex. This is an argument for only using impostors of the same sex as the client.

Nevertheless , cross gender confusion does occur and must be guarded against. For this reason both impostor and clients sets should contain both male and female speakers, preferably in equal numbers.

Note that tasks where only same-sex impostors are used will have error rates approaching twice that for tasks using both genders, since the same-sex speakers cause most of the errors.

Dialect and Accent

Dialect and accent is intuitively one of the key features used by human listeners for speaker recognition. It seems likely that discriminating speakers with the same dialect and accent will be more difficult than discriminating speakers with different dialects and accents.

Many speaker recognition databases contain a wide range of dialects and accents in order to accurately represent the intended client base of the system. That is as it should be, but it must be noted that if clients and impostors in a database have the same dialect or accent, then the error rates will be higher than if the dialect or accent differs. This is another factor which makes comparisons of systems which use different databases extremely difficult.

2.3.4 Isolated Words and Connected Speech

Data consisting of isolated words contains no inter-word co-articulation effects, while the acoustic realisation of a word in connected speech will be influenced by the words immediately before

and after it. Constraining the task so that the speech consists of isolated words therefore allows more reliable word models to be constructed than would be possible using connected speech.

Isolated words can occur in many potential applications. Any single word password or menu-driven spoken command task will contain isolated words. If necessary users can be forced, by the use of separating tones, to provide isolated words where they might otherwise produce connected speech. This is, however, rather unnatural and clumsy, and if possible isolated words should only be used for tasks where they would occur naturally.

Isolated words are used for the experiments in this thesis because of the limited amount of connected digit data available.

In general an isolated word task will be less difficult than a connected speech task because connected speech is more variable. Some quantification of this difference can be gained from the results of (Rosenberg *et al.*, 1990b; Rosenberg *et al.*, 1991).

2.3.5 Data Storage and Computation Restrictions

In practice, limitations on data storage and computation are not generally important elements in the design of an ASV system. Real-time performance is much more easily achievable for ASV than for ASR, which has associated grammar search requirements. Nevertheless, the feasibility of real time operation must be kept in mind, and some cohort normalisation schemes[4] for ASV which involve applying the bid utterance to a large number of models would be difficult to implement cheaply on current technology.

Data storage requirements are only likely to be an issue in two cases

1. When a client's models must be portable, such as a system using magnetic or smart cards, where the client model is encoded on the card.

2. When the client population is very large, such that the overall storage requirements of the client database are significant compared to typical hard-disk capacities.

These two design conditions can be assessed in a binary, rather than a continuous manner. They are either satisfied or not. It is unlikely, for instance, that two real-time systems would be subject to market differentiation on the basis of computational speed.

[4] Refer to Section 2.4.4.

2.3.6 Training Requirements

For any given task involving statistical models it is desirable to use as much data as can be obtained. The nature of speaker recognition is such that training data must be obtained from each and every client speaker. The amount of training data available for training the client models is, therefore, strictly limited by what the client will find acceptable. Further data can be obtained to adapt and improve the initial models while the system is in use, but the initial performance must be acceptable.

The amount of speech data that a client will willingly provide depends on the application and the potential benefits that are available. The amount of training data that is available is therefore task-dependent. In physical access control applications such as door entry systems, a client is likely to be willing to invest significant time and effort in order to ensure ease of use and a high level of security. For applications such as telephone banking where a client is replacing an existing service with a more *convenient* one, the amount of *inconvenience* associated with obtaining the new service will be critical to its success.

Although companies interested in speaker recognition applications are attempting to quantify the amount of data that can be realistically demanded, such information is not available in the literature. Five utterances of each of the digits is a common amount when researchers are attempting to be realistic about training data quantities. This is probably based more on the minimum amount of data that can be used to train a reliable HMM than any assessment of client tolerance. Five tokens of each of the 10 digits would take a minimum of one minute to collect, which is probably reasonable.

The performance of ASV systems varies strongly with the amount of training data used and comparison of different systems which use different amounts of training data are not very meaningful. When comparing systems it is important to consider not just the amount of data used to train the models but also the number of sessions which were used to collect it. Ideally several sessions, spaced over a period of weeks or months should be used so that temporal variations in the speakers voice are well modelled. It is unlikely, however, that more than one training session could be expected in a telephone banking application, and certainly no more than two. Clients will be enrolling because they want to use the system - they will not want to wait several days for it to become available. It is very difficult to compare systems where multiple enrolment sessions are used to a system where a single enrolment session is used.

2.4 Automatic Speaker Verification Systems

The basic components of an ASV system are shown in Figure 2.3 and are discussed in the following sections.

2.4.1 Feature Extraction

The purpose of feature extraction is to condense and distill the important information in the speech signal. Any source of variability which is not important to the task should ideally be suppressed or eliminated. For the ASV task, the information that is relevant is information about the speaker. This includes both physical information about the size and shape of their vocal apparatus and also behavioural information such as accent and speaking rate. Ideally the information that the features emphasise should have small intra-speaker variation and large inter-speaker variation. They should be easily extracted and not change over time or be affected by the speaker's health or emotional state. They should also be robust to varying channel characteristics and not be consciously modifiable by the speaker, so that it is difficult for an impostor to disguise their voice. This is a very demanding set of requirements.

A starting point in developing a good feature set is to study which aspects of the speech signal are important to human perception of voice individuality. Experiments in (Itoh & Saito, 1982) which are discussed in (Furui, 1986) used analysis-resynthesis to determine the effect of spectral envelope, pitch and dynamic (durational) characteristics on the perception of voice individuality. It was found that the spectral envelope information dominates the pitch and dynamic characteristics, which only become important if the spectral envelope information is missing.

For most speech features the assumption is made that the speech signal is stationary for periods of 10-45ms and overlapping windows of that size are applied to the speech signal. This allows a short term spectrum to be computed for each window or *frame* of speech. The reader is referred to the excellent text by (Rabiner & Hwang, 1992) for details of standard pre-processing techniques.

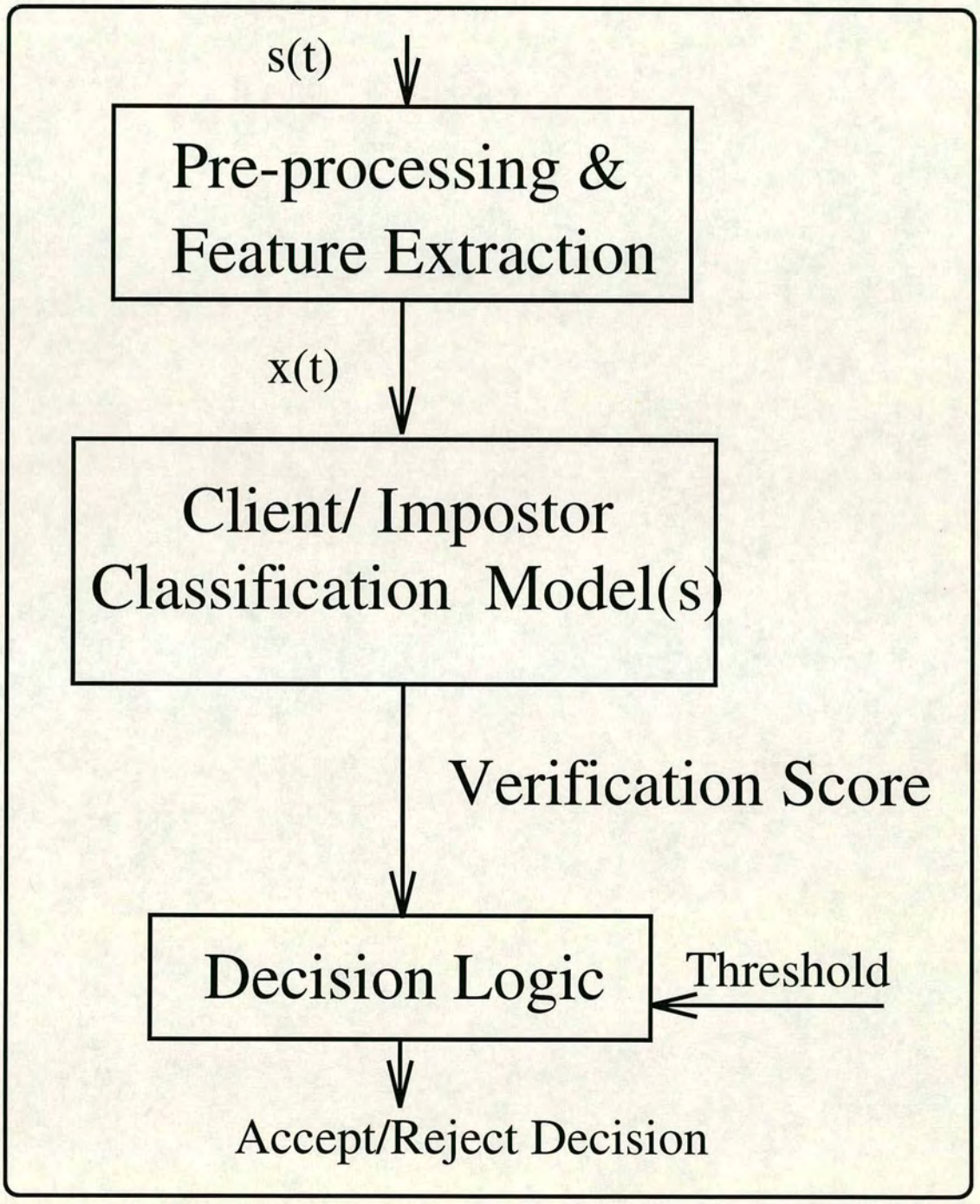

Figure 2.3: Block diagram of a generic ASV system.

Pitch

Pitch is an intuitive feature to use for speaker recognition but experimental studies have shown varying results. Pitch was heavily used in early systems (Sutherland & Jack, 1988), but its susceptibility to mimicry and its large intra-speaker variation with mood and health of the speaker (Rosenberg, 1976) have meant that its usefulness outside the laboratory has been limited. Yegnanarayana, however, used prosodic features of F0 contour and duration for speaker recognition in Hindi in a recent study (Yegnanarayana *et al.*, 1994).

LPC Cepstral Coefficients

The cepstrum of a signal is the Fourier transform of the log of the spectrum[5]. The cepstrum's ability to capture the formant structure and spectral tilt of the windowed speech segment has made it the most common choice of feature for speech technology applications.

Atal (Atal, 1976) compared LPC predictor coefficients, autocorrelation coefficients and LPC cepstral coefficients and found that the LPC cepstra feature set was the best feature set when used with a Mahalanobis distance measure for speaker recognition. Furui (Furui, 1981) found cepstra to be better than a log area ratio representation.

There have been many suggestions for improvements to the cepstrum. For telephone channels in particular the subtraction of the cepstral mean from each cepstral vector can help combat the effects of additive noise, which appears as a bias in the cepstral domain (Rosenberg *et al.*, 1994). However, Reynolds reports that this technique reduces performance when there is not much channel variation (Reynolds, 1994).

A perceptually based frequency scale known as the mel-scale has been determined experimentally. The mel is based on the ear's ability to distinguish one frequency from an adjacent one. For higher frequencies the ear's resolving power is reduced. The mel scale reflects this by using a non-linear mapping from Hz to mels, giving less frequency resolution as frequency increases. Mel frequency cepstral coefficients (MFCC) are commonly used in speaker recognition systems (Carey & Parris, 1992; Gish, 1990; Openshaw *et al.*, 1993; Rose & Renolds, 1990). The reasoning is that the use of the mel frequency scale will emphasise perceptually important aspects of the speech signal. The use of a mel frequency scale has the effect of emphasising the

[5]The name cepstrum comes from the fact that *ceps* is *spec* backwards.

lower frequencies.

A technique called adaptive component weighting (ACW) for cepstral coefficients has recently been proposed to produce frame-dependent weights which emphasise the formant structure in the speech signal (Assaleh & Mammone, 1994).

Delta (difference) Coefficients

The use of difference or Δcepstra as a feature set is widespread in both ASR and ASV systems. The difference cepstra is a first order approximation to the first differential of the cepstra. The first order finite differential is intrinsically noisy and the use of an orthogonal polynomial fit of each cepstral coefficient trajectory over a finite window was proposed in (Furui, 1981). Furui also showed that a first order polynomial was sufficient.

The motivation behind using the difference cepstra is to capture the transitional, as opposed to the instantaneous nature of the spectrum. An important paper on the combined use of cepstra and Δcepstra for speaker recognition is (Soong & Rosenberg, 1988). Soong used a vector quantisation (VQ) codebook system on a TI ASI task to evaluate the feature sets. Soong normalised the instantaneous and transitional distances by dividing by their standard deviations. The correlation coefficient of the two distances was 0.6, which considering that they are both representations of the same speech data indicates significant independence, and that the two distances could be usefully combined.

Two different approaches have been taken to combining the use of cepstra and Δcepstra feature sets. Rosenberg (Rosenberg *et al.*, 1990b) concatenates the 12 cepstra and 12 Δcepstra coefficients to form a 24 dimensional feature vector. Concatenating the feature vectors has two effects - the relative weighting of the cepstra and Δ cepstra information is fixed at the start of the modelling process[6] and the dimensionality of the feature space is increased dramatically. In designing the HASAS system it was argued that neither of these effects is desirable and the two feature sets were kept separate, allowing their relative importance to be adjusted at the verification stage. This is consistent to the approach used by Soong who used two VQ codebooks, one for cepstra and one for Δcepstra and combined the results from the two codebooks in a linear weighted sum. Soong found that the instantaneous features performed better than the transitional

[6]The relative weighting is determined by the covariance of the Gaussian mixture models.

features, but that the combination of both features performed better still. When a spectral tilt was added to the test data to create an artificial channel mismatch between training and testing data, the transitional feature performance was not affected, while the error rate of the instantaneous features was increased around 50%.

Interestingly it was noted that the system was much more susceptible to cross-sex confusion when transitional features were used than when instantaneous features were used[7].

Soong suggests that the benefits of transitional features are greater for a TI task than a TD task since the temporal alignment of TD approaches directly accounts for temporal and contextual information. This was supported by a TD experiment in which the improvement in single digit ASI error rate gained from using the combination of instantaneous and transitional features was roughly a third that which was gained on the TI task.

New Features

An alternative perceptually based feature set to the MFCC is the popular perceptual linear predication (PLP) features, usually combined with RASTA (RelAtive SpecTrAl) processing as RASTA-PLP (Koehler *et al.*, 1994; Hermansky *et al.*, 1991). This feature set has been successfully used for ASV (Rajasekaran, 1993) and is compared with, and combined with MFCC for ASV in (Openshaw *et al.*, 1993). Another feature set recently used for ASV is line spectral pairs (LSP) (Yuan *et al.*, 1993).

2.4.2 Modelling Techniques

The main modelling techniques currently used for ASV are described in this section.

Long Term Statistics

Text-independent ASV is a more difficult task than text-dependent ASV. The lack of any constraint on what is said makes it impossible to model specific words. If only a few commonly occurring phones are modelled, much of the information in the test speech is wasted and obtaining a complete set of phone models for each client would require too much training data.

[7] Cross-sex confusions were 26.4% of total speaker confusions for the transitional features compared to 9.4% for instantaneous features.

ASV algorithms based on short or long term statistics of speech features are commonly used for TI ASV tasks because they are computationally fast, easy to implement and surprisingly robust.

The approach of Gish (Gish *et al.*, 1994) is based on segmental statistics of speech features and has various innovations to improve robustness. Much of this robustness is due to combining as many different information sources as possible. Several statistics are used -the mean and covariance of the cepstral coefficients and the covariance of the cepstral derivatives. Frames are energy filtered so that only the high energy frames are used and the scores from this approach are combined with the scores obtained when energy filtering is not used. Segment scores are normalised using a log likelihood ratio and the worst scores are pruned on the assumption that they come from contaminated segments.

This combination of several robustness strategies is very successful. Evaluated on the SWITCHBOARD telephone-speech database (Godfrey *et al.*, 1992), the system produced no speaker identification errors on a 24 speaker task, compared to the 4% error rate obtained by (Higgins *et al.*, 1993) on the same task.

While this is an excellent result, six 60-second training sessions were used to construct client models and 45 second recordings containing around 30 seconds of actual speech were used for testing. The text-dependent task that HASAS is designed for has much stronger constraints on enrolment and test data.

Vector Quantisation

Another technique commonly used for TI tasks is that of vector quantisation (VQ) codebooks. Numerous VQ codebook approaches have been proposed for ASV and ASI (Matsui & Furui, 1994b; Matsui & Furui, 1991; Rosenberg & Soong, 1987). A codebook is a group of points in a feature space. The distribution of the points or *codewords* has been designed in order to cover the useful feature space. One way to construct a codebook is to cluster a large number of feature vectors and to use the means of these clusters as codewords. The essence of the technique for ASV is that if the codebook is speaker specific, the positions of the codewords represent a model of the client's speech in the feature space. When testing, each feature vector is compared to the codewords and some form of distance between the feature vector and the nearest codeword or codewords is calculated. The average distance over all feature vectors can then be used as a verification or identification score. There are many variations which are possible. Frames

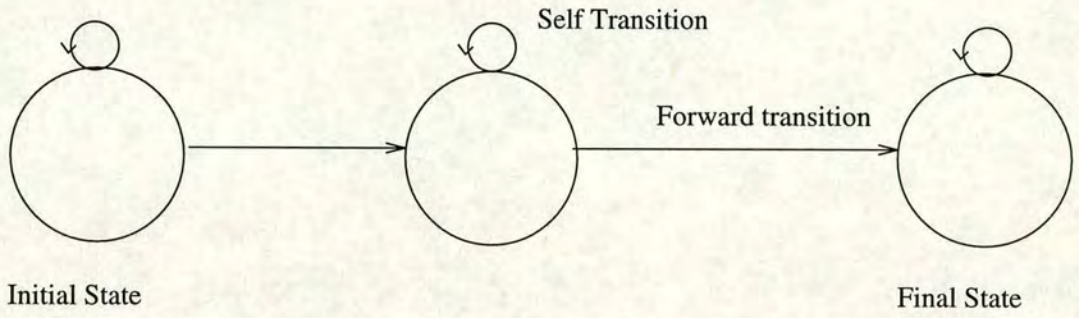

Figure 2.4: Schematic of a basic 3 state left-to-right HMM.

can be selected according to some criteria, so that only frames which are particularly speaker discriminating are used (silence frames, for instance, should not be used). The weakness of the codebook approach is that all frames are treated independently and no sequential information is used. This makes codebook approaches more suited to TI tasks than TD tasks.

Template-based approaches

Dynamic time warping (DTW) and other template-matching techniques have been generally superseded by the more powerful and flexible hidden Markov models (HMM), and although research interest still remains (Hannah *et al.*, 1994; Hannah *et al.*, 1993) it has slowed dramatically. DTW remains a useful technique when there is insufficient data to train an HMM. In (Rosenberg *et al.*, 1991) whole-word continuous HMMs (CHMM) were shown to reduce the EER by 50% compared to a template-based system applied to the same data. DTW is also part of the continuum from codebooks to HMMs.

Hidden Markov Models

Hidden Markov model (HMM) theory and application is now the core of speech and speaker recognition research. Numerous references are available to readers unacquainted with HMM theory (Huang *et al.*, 1990; Rabiner & Hwang, 1992).

Basically HMMs improve on simple VQ codebook approaches by providing a probabilistic framework for modelling temporal and contextual information. An HMM consists of a number of states, each state having its own mapping of the feature space into an observation probability space. The observation probability for a feature vector therefore depends on the state.

In a strict left to right HMM as shown in Figure 2.4 the constraint is imposed that the first frame of speech is allocated to the first state of the model and the last frame of speech is allocated to the last frame of the model. A strict left to right HMM structure is used in this work and the following discussion assumes such a structure. The Viterbi dynamic programming algorithm can be used to find the optimal allocation of frames to states, based on maximising the total probability. This process is termed the state segmentation. There are two possibilities for each frame. The frame can either be allocated to the same state as the frame before (a self transition) or it can be allocated to the next state (a transition to the next state). In order to improve the modelling of the temporal structure of the speech signal, each state has probabilities for these two types of transition. These are known as transition probabilities. The two transition probabilities sum to one. The Viterbi algorithm produces a state segmentation which maximises the product of all the observation probabilities and all the transition probabilities. The total probability of the optimal state segmentation can be used as a measure of the match between the model and the speech.

The transition probabilities can be replaced by state duration probabilities to improve the modelling of temporal structure, although this makes the state segmentation more difficult.

The mappings from feature space to observation probability space in each state are optimised in some way during training. The most common optimisation criterion is to maximise the total probability over all the training utterances for all possible state sequences. This can be done efficiently using the Baum-Welch algorithm (Baum *et al.*, 1970).

The mappings have several different forms. Discrete HMMs (DHMM) make use of VQ codebooks. Each codeword in the codebook is assigned a probability and the probability of the codeword which is closest to the feature vector is used as the observation probability. This leads to a very quantised observation probability space. If the codebook size is 256 then only 256 different values of observation probability are possible. To overcome quantisation error, continuous HMMs (CHMM) can be used. In CHMMs each state has a probability density function (PDF), which is typically a mixture of Gaussian functions. The combination of the probabilities from these Gaussians provides the mapping from feature space to observation probability space. The parameters of the Gaussians must, however, be estimated for each state of each model, and this can be difficult when the amount of training data is small. Semi-continuous HMMs (SCHMM) overcome this problem by having a fixed set of Gaussians in

a codebook that are shared for all states. Each state simply trains different weights for each Gaussian. If necessary, the means of the Gaussians can also be re-estimated.

A comparative study of HMM and VQ systems for a text-dependent task can be found in (Rosenberg *et al.*, 1990b). The speaker specific EER for a 7 digit string was 2.9% for the VQ system, compared to 1.8% for the HMM system. It would be expected that HMMs would out-perform VQ in text-dependent tasks because of their ability to model temporal structure. All frames are treated independently in a VQ system, whereas an HMM makes use of the temporal contextual structure of the utterance. HMMs are now the predominant modelling technique for TD ASV and their use is discussed in more detail in Section 2.4.3.

HMMs can be used for TI tasks by collapsing the state structure and using single state HMMs. This is the approach taken by Reynolds for his Gaussian mixture models (GMM) (Reynolds, 1994). On the SWITCHBOARD database (Godfrey *et al.*, 1992) using 3 minutes of enrolment speech and 10 seconds of test speech on a TI task a 7% SI EER was obtained (Reynolds, 1994).

Neural Networks

Artificial neural networks (NN) are a powerful and flexible architecture for solving classification problems. They are easy to implement and, more importantly, they are well suited to discriminative training. It is no surprise, then, that various NN approaches have been tried for the ASV task[8].

The discriminating power of neural nets is their main advantage over HMMs. For speech applications, however, neural nets have so far been unable to match the performance of HMM based systems because of the HMM's superior modelling of temporal structure. For this reason NN have been most successful in text-independent tasks (Farrell *et al.*, 1994; Hattori, 1994; Oglesby & Mason, 1990; Oglesby & Mason, 1991) where, in general, all frames are treated independently and non-sequentially. Neural networks have produced results comparable to those using VQ codebook techniques for text-independent tasks (Farrell & Mammone, 1994; Oglesby & Mason, 1990). This makes sense since VQ codebooks are similar to single state DHMMs which lack the temporal structure implicit in left-to-right state transitions.

The relative strengths of NN and HMM approaches have stimulated much research. Time

[8] For recent reviews see (Tsoi *et al.*, 1994; Bennani & Gallinari, 1994)

delay neural nets (TDNN) and recurrent neural nets (RNN) (Tsoi *et al.*, 1994) have been investigated as a way to improve the neural net's use of contextual information. Neural nets have been used in parallel with HMMs as a second classifier in a speech recogniser (Devillers & Dugast, 1993). Perhaps the most important development has been the use of hybrid HMM/NN architectures (Naik & Lubensky, 1994; Bridle, 1990). Hybrid HMM/NN approaches aim to combine the best features of both architectures and have proven very successful in ASR tasks (Hochberg *et al.*, 1995; Bourlard & Morgan, 1994). Their application is a promising area of current research.

Model Adaptation

Model adaptation is often used in speech recognition systems to improve models trained on a small amount of training data and to turn speaker independent models into speaker dependent models in order to improve performance (Schiel, 1993). The concluding sentence of a PhD thesis on the adaptation of reference patterns for word-based ASR (McInnes, 1988) is as follows.

> In general, adaptation of reference patterns is a valuable enhancement to a speech recognition system, especially where the speech encountered as recognition input is expected to differ systematically in some respect from the training speech, or to exhibit a drift over time, or where it is inconvenient to use an extensive initial training procedure.

The ASV task fits this description perfectly. The work of Furui (Furui, 1986) shows that client speech will vary significantly over both the short and long term, and almost all applications will find extensive initial training to be unrealistic.

In order to improve sparsely trained initial models and also to model temporal drift in the client's voice, model adaptation over time will be an essential part of any ASV system. Experiments have been done into the effect of adaptation on error rate (Rosenberg *et al.*, 1990b; Rosenberg *et al.*, 1992) and the improvement in performance makes it clear that the intra-speaker variability which is introduced by the passage of time is more than compensated for by the increasing robustness of the adapted models. This means that the steady state performance of the system will exceed the initial performance. This work therefore concentrates on maximising initial performance with the assumption that speaker adaptation would be part of any commercial

implementation.

2.4.3 Hidden Markov Model based ASV Systems

HMMs can be, and have been, applied to the ASV task in many ways. Firstly there is the issue of which architecture to use. The option of explicit state duration modelling must be considered, along with the use of silence and noise models. A choice must be made whether to use whole or sub-word modelling units, and how many mixture components or codewords to use and how many states each model should have.

Every issue has been studied to some extent in the literature, but many choices still involve trade-offs which are dependent on the specific task being considered. This section discusses these design issues, with reference to several recent systems reported in the literature. Chapter 3 outlines the design choices which were made in the specification of HASAS.

HMM Architecture

The different HMM architectures represent different points on a continuum of feature space representations. From one point of view DHMM represent one end of the scale and CHMM the other with SCHMM in between. Using fuzzy VQ, or VQ with distance or distortion measures, makes them more like SCHMM and tying the mixtures in CHMM makes *them* more like SCHMM. While the terms DHMM, CHMM and SCHMM all still appear in the literature, closer examination reveals that most model topologies now occupy the middle ground. The reason for this is that the middle ground represents a good compromise given the limited training data available for ASV. Both SCHMM (Forsyth & Jack, 1993) and CHMM (Matsui & Furui, 1992a) have been shown to be superior to DHMM for speaker recognition.

Ergodic or Left-to-Right HMM?

The first studies to use HMMs for ASV used ergodic HMMs[9] (Poritz, 1982; Tishby, 1991). Tishby reported only a slight improvement over VQ in using ergodic HMMs. Rosenberg used a left-to-right architecture[10] and compared his results with those of Tishby, who used the same

[9] Transitions between any two states are allowed.
[10] Once a state has been exited it cannot be revisited.

database. The left-to-right architecture was found to be superior[11], and this was attributed to the left-to-right structure providing a more rigid temporal structure (Rosenberg *et al.*, 1990b). Left-to-right models are used in HASAS.

Whole-word or Sub-word Models?

Sub-word units are widely used in ASR systems to improve performance, but that is for large vocabulary systems, where a large amount of training data is available. Most ASV systems use small vocabularies and very limited training data. The strict limit on training data prevents the use of context-dependent sub-word models, which improve performance on connected speech tasks by modelling co-articulation effects. If sub-word units are used they must be context-independent, at least to start with, and they therefore provide no advantage over whole word models.

In applications with a limited vocabulary of isolated digits, as is the case with HASAS, there are no inter-word co-articulation effects, and the intra-word co-articulation effects will be well modelled by whole-word models. For this reason, whole-word models are used in HASAS.

There are two reasons to consider the use of sub-word models instead of whole-word models. Firstly, if a complete set of sub-word units is available for each speaker, randomly prompted passwords or phrases can be generated which have never been used before by the client speaker. This means that recordings cannot be used to defeat the system. The second advantage is that the system can be either text-independent or text-dependent, which makes it more versatile.

The vocabulary of the YOHO database (Campbell, 1995), consists of combination-lock phrases, such as *eighty-two*. This is an example where the vocabulary can be expanded by clever use of sub-word units. In this case the use of sub-word units is desirable, because 80 digit-combinations (20-99) can be constructed using the models of the nine digits (1-9) plus *twen, thir, fif* and *ty*. The high frequency of the *ty* syllable allows it to be modelled in several contexts.

In summary the advantage of sub-word models comes not from a performance advantage, but in providing a flexible vocabulary and possibly text-independent capability. The feasibility of using sub-word units depends on how realistic it is to require clients to provide sufficient

[11] 1.1% rather than 2.0% speaker specific EER. 7 digit test utterance, using 10 training tokens.

training data at enrolment, and whether the initial performance of the context-independent models is sufficiently high.

Duration Modelling

Tishby (Tishby, 1991) found that the transition probabilities do contain some speaker discriminating information but that their discriminating power was poor compared to the observation probabilities and proposed that explicit state duration modelling should improve speaker verification performance. State duration modelling has been used successfully in ASR (Levinson, 1986).

The value of Gaussian state duration modelling for ASV is investigated in Section 4.8 where it is found that while the state duration modelling does contain speaker discriminating information, it does not provide any additional useful information when two spectrally based feature sets are used.

If the state duration probabilities provide no additional speaker discriminating information, then fixed transition probabilities could well be detrimental to the verification decision. The first reason for this is that, because they are not an accurate model of state duration, the information they contain will not be as reliable. The main reason, however, is that their weighting relative to the observation probabilities is fixed. In a traditional verification score there are as many transition probabilities as there are observation probabilities[12], yet as Tishby found, and the experiments in Section 4.8 confirm, the spectrally-based observation probabilities are far more important. An equal weighting of transition probabilities and observation probabilities is therefore likely to produce worse performance than leaving the transition probabilities out of the verification score altogether.

In Rosenberg's HMM system (Rosenberg *et al.*, 1992) all transition probabilities are made equal, thereby neutralising their effect. This is probably detrimental to the state segmentation, however, and it perhaps would be better to simply leave the transition probabilities out of the calculation of the verification score[13]. Rosenberg uses a word duration model to produce a duration probability which is added to the verification score. The effect of this was not reported.

[12] Although they do have less dynamic range (Tishby, 1991).

[13] By using the verification score calculation given in Equation 3.23

Silence and Noise Models

It is beneficial to explicitly model silence and noise. Reynolds uses an energy-based speech activity detector (SAD) to discard silence/noise frames in his text-independent ASV system (Reynolds, 1992). Rosenberg uses a 1-state silence model and a 3-state artifact model trained from speaker generated puffs and clicks. Noise and silence models are used before and after words. There is no explicit noise modelling during words (Rosenberg *et al.*, 1992).

Discriminative Training

The most common form of training for HMMs is maximum likelihood estimation (MLE) in which the model estimation is based on maximising the likelihood of the training data over all training utterances. Several forms of *discriminative* training have been proposed for HMM based ASR, such as maximum mutual information estimation (MMIE) (Bahl *et al.*, 1986; Normandin *et al.*, 1994) and minimum discrimination information (Epraim & Rabiner, 1988)

Recently Liu (Liu *et al.*, 1994) proposed a minimum classification error (MCE) approach to discriminative training for speaker recognition tasks which uses a variation of the generalised probabilistic descent (GPD) algorithm (Chou *et al.*, 1993; Chou *et al.*, 1992) to estimate model parameters. This training approach attempts to minimise the recognition error on the training data by taking into account competing models from other speakers.

The improvements gained from the discriminating training were modest. The single digit SS EER was 1.07% using MLE models and 0.81% using MCE trained models. This is a reduction of only 24% and it was reported that the amount of improvement dropped as the number of digits in the test utterance was increased. Speaker normalisation (discussed in Section 2.4.4) was used on both MLE and MCE trained models.

Although this study does show that discriminative training offers an improvement over MLE training for ASV, the benefits of discriminative training are less than might be expected. The success of DOP modelling detailed in Chapter 5 offers some insight into this. In text-dependent verification there are two processes going on, speech recognition and speaker recognition. The state segmentation is a speech recognition process and the verification score calculation is a speaker recognition process. The likelihood score of a speaker dependent model is both a speech and a speaker recognition score. MCE training maximises the model's *speaker* recognition

performance on the training data. It is possible that this decreases the model's *speech* recognition performance on unseen data. The problem with using conventional HMMs for ASV is that they are both speech and speaker recognition models and the verification score has both speech and speaker recognition components. MCE training shifts the balance more towards speaker modelling than speech modelling, but the problem still remains. The success of the DOP models lies in the way that they separate speech and speaker modelling, as we shall see.

2.4.4 Speaker Normalisation

Since the publication by Higgins (Higgins *et al.*, 1991), the use of speaker normalisation techniques has become widespread (Matsui & Furui, 1992b; Reynolds, 1994; Rosenberg *et al.*, 1992; de Veth *et al.*, 1993). Several variations have been proposed, based around the same principle.

The problem that these methods aim to address is that of channel variation, particularly handset variation in telephone applications. The normalisation technique can be applied to any modelling technique which produces some form of likelihood score.

Assume for the sake of illustration that the verification score from the speaker dependent model V_{SD} consists of two parts, a *speech* recognition score V_{speech}, and a *speaker* recognition score $V_{speaker}$.

$$V_{sd} = V_{speech} \times V_{speaker} \tag{2.4}$$

Now $V_{speaker}$ is a measure of how likely it is that the utterance came from the modelled speaker, which is the desired quantity for speaker verification. V_{speech} is a measure of how likely it is that the correct text was uttered.

When the task is ASR, $V_{speaker}$ is an unknown variable which prevents V_{SD} being a good approximation to V_{speech}. When the task is ASV the reverse is true and V_{speech} is an unknown variable which prevents V_{SD} being a good approximation to $V_{speaker}$. Fortunately, for a given speaker, channel, and text, the value of V_{speech} does not vary much and a reliable threshold can be set that assumes a relatively constant V_{speech}.

Unfortunately for telephone applications the channel varies considerably and V_{speech} is sensitive to the channel. Thus for a given speaker and utterance there is considerable variation in V_{speech}. Also V_{speech} varies according to the text and it is often required that a threshold be used that is independent of the text.

If V_{speech} is not constant, then V_{SD} is not a reliable estimate of $V_{speaker}$. The idea behind

speaker normalisation techniques is to use an estimate of V_{speech} (labelled V'_{speech}) to get a better estimate of $V_{speaker}$ (labelled $V'_{speaker}$) as shown in Equation 2.5 (derived from Equation 2.4).

$$\log(V_{SD}) - \log(V'_{speech}) = \log(V'_{speaker}) \tag{2.5}$$

V'_{speech} can be found by applying the utterance to one or more *anti-speaker or impostor* models.

Choosing an Imposter Model

The impostor model or models (λ_I) can be constructed in a variety of ways. Some of the methods which have been proposed for the construction of a impostor model are discussed in this section.

- A single model of all speakers (a speaker independent model) as used by (Carey & Parris, 1992) (Matsui & Furui, 1994a).
- The mean or some other statistic of the log likelihood score from a large group of speaker dependent models. The models are of speakers chosen at random and we will refer to them as the *random anti-speaker set*. This set does *not* include the client speaker (Higgins *et al.*, 1991) and (Reynolds, 1994).
- The sum of the likelihood scores of a large group of speaker dependent models, which *does* include the client speaker. This method is based on the *a posteriori* probability (Furui, 1994; Matsui & Furui, 1994a)
- The mean or some other statistic of the log likelihood scores from a small group of speaker dependent models. The models are of speakers who have been selected as being similar to the client speaker. This is known as the *cohort speaker* approach, and is described in (Rosenberg *et al.*, 1992).

No study has yet shown clear performance advantage of one form over all others. Such a study must be conducted carefully because there are sources of experimental bias associated with cohort speaker techniques which will be discussed in the following section.

Experimental Bias in Cohort Speaker Normalisation Experiments

There is a source of experimental bias associated with cohort speaker techniques if the pool of cohort speakers is not completely independent from the set of impostor speakers. Since the cohort speakers are explicitly modelled as impostors it is not realistic to include them in the impostor population, as they are *closed test* speakers and would almost certainly be correctly rejected. We will call this the experimentally-invalid-impostor (EII) bias. What is done in most studies to avoid EII bias is to leave out the cohort speakers for each client from the impostor set for that client. The problem with this is that in eliminating one source of experimental bias, another bias is created which has often been overlooked, but which is which is just as significant.

The problem arises because the k cohort speakers used for client A are, by definition, the k speakers in the database who are most likely to be successful impostors against client A. The elimination of the cohort speakers eliminates the most similar client/impostor pairs, which probably means most of the error-generating match-ups, thereby creating a significant experimental bias. We call this bias the eliminating-best-impostors (EBI) bias.

In the experiments comparing normalised results with un-normalised results, Rosenberg allows for both EII and EBI bias by removing the cohort speakers from the impostor set for the experiments *without* normalisation as well as for the experiments *with* normalisation. In one set of experiments[14] the removal of the cohort speakers from the impostor set produced a reduction in the EER from 4.7% to 2.9% using un-normalised models. By adding normalisation the EER was reduced to 2.6%. The removal of good impostors clearly has a more dominant effect than the use of normalisation.

Two sources of experimental bias have been discussed so far. Both EII and EBI bias can be eliminated by using the experimental method used by de Veth (de Veth *et al.*, 1993) in which the best cohort score is left out for each utterance. The assumption is that the best cohort score for an utterance from impostor I_j will generally be from the model of impostor I_j. Eliminating that score eliminates that impostor model from the cohort set, which eliminates EII bias, but does not eliminate that utterance from the impostor data set, and so avoiding EBI bias.

This experimental method of de Veth is better than other cohort speaker studies, but it too has a flaw. Eliminating the best cohort score avoids the first two sources of bias but creates a

[14] Where there was no handset mismatch.

third bias, which we will call the unbalanced-normalisation (UN) bias. De Veth uses the average of the best four cohort scores for normalisation. In the case of impostor utterances, removing the best cohort score causes the best four non-biased scores to be used, which is the desired effect. In the case of client utterances, however, removing the best cohort score has the effect of reducing the average of the best four cohort scores which in turn has the effect of increasing the normalised score. Increasing the normalised scores of client utterances but not of impostor utterances will improve performance and so creates UN bias. To avoid UN bias, the best cohort score should really only be removed in the case of impostor utterances.

Calculating the Cohort Impostor Score

Rosenberg (Rosenberg *et al.*, 1992) found that using the mean of the log cohort scores was better than using the maximum, median or 80^{th} percentile. This was for the use of best-match cohorts. De Veth (de Veth *et al.*, 1993) normalises by subtracting the mean log cohort score and then dividing by the standard deviation of the log cohort scores. No direct comparison between these two approaches has been made.

Cohort Selection

Rosenberg selects cohorts by matching cohort training data with client models and matching client training data with cohort models, then taking the best combined results. Chen (Chen *et al.*, 1994) has a similar approach but only matches client data with cohort models. While this is less intuitive than matching cohort data with client models, since that is what will happen during testing, it has the benefit that the cohort models are available in a practical system, whereas it is less convenient to have cohort speech available during the enrolment of each client speaker. De Veth (de Veth *et al.*, 1993) doesn't use a cohort set in the same way as Rosenberg. A fixed anti-speaker set is used instead for all speakers but only the anti-speaker models producing the N-best scores are used as cohort scores. This has the disadvantage that decreasing N does not decrease the amount of computation required, but it has the advantage that no cohort selection procedure is required at enrolment time.

If cohort speakers are used the question of how many speakers to use arises. Once again, experimental bias has clouded the issue. Figure 2 in (Rosenberg *et al.*, 1992) shows that

performance improves as the number of cohort speakers k is increased from one to five. This result can possibly be explained in terms of increasing EBI bias as more and more good impostors are being excluded from the impostor set. The apparent levelling off around k = 5 can also be explained purely in terms of EBI bias. When k is increased to the point where the the cohort speakers who are being added are not successful impostors, their removal from the impostor set does not reduce the EER, thereby artificially creating some apparently optimal number of cohort speakers, but the point at which levelling off occurs is actually just related to the number of good impostors in the database. De Veth avoids the EBI bias and found very little variation in performance as the cohort size increases.

An alternative approach to cohorts is the use of a random anti-speaker set. EBI bias also affects comparisons of cohort speakers with a randomly chosen anti-speaker set. In (Reynolds, 1994) a series of experiments comparing cohorts with randomly chosen anti-speakers can be well explained in terms of EBI bias. Although it is not explicitly stated, we can assume that the cohort speakers are being excluded from the impostor set[15]. The results[16] show a decrease in EER as the number of cohort speakers increases, but that if more than two anti-speakers are used, the size of the anti-speaker set does not effect the EER. Given a large enough sample size, the randomly chosen anti-speakers will have roughly the same proportion of *good* impostors as the total impostor population, so eliminating them from the impostor set should not have as great an effect on the EER[17] as removing the k *best* impostors.

An interesting result from (Reynolds, 1994) is that for any number of anti-speakers from two to twelve, randomly chosen sets perform better than cohorts sets. Note that this is even more significant given the EBI bias in *favour* of cohorts. Reynolds explains this as cohorts making the system vulnerable to FA errors from impostors who do not match either the client or the impostor model well, such as opposite sex impostors. Reynolds suggests that opposite sex impostors could be excluded by a separate method. Subsequent work has shown that the vulnerability can be easily eliminated using a preliminary rough classification to eliminate obvious impostors (Chen *et al.*, 1994). This approach is probably unnecessarily complicated, since applying a weak threshold to the client model score before normalisation would be a simple way to achieve the

[15] If they are not then the source of bias will be EII bias but the analysis is still valid

[16] Figure 4 of (Reynolds, 1994)

[17] We assume that any effect that a non-representative random sample of cohort speakers might have will not be significant.

same effect.

These results which favour random selection of anti-speakers appear to conflict with earlier results in (Rosenberg *et al.*, 1992) from which it was concluded that random anti-speakers were not as useful as cohorts. In Rosenberg's experiments the EER with the full impostor set was 4.7%[18]. Using randomly chosen anti-speakers (and removing them from the impostor set) the EER was reduced to 3.5%. This is an unbiased comparison because the impostors removed were randomly chosen. When cohort speakers were removed from the impostor set the un-normalised EER was 2.9% while the normalised EER was 2.6%. The decision on whether best-match cohorts are better than randomly chosen cohorts should be based on whether the reduction from 2.9% to 2.6% is better than a reduction from 4.7% to 3.5% rather than whether 2.6% is better than 3.5%. Interpreting the results in this way leads to the conclusion that randomly chosen anti-speakers are better than cohort speakers, which agrees with, rather than opposes the findings of (Reynolds, 1994).

On balance the evidence appears to suggest that anti-speakers should be randomly chosen rather than selecting a cohort. If random anti-speakers are used it is logical to use the largest anti-speaker set that is computationally feasible in order to get a robust average.

Speaker Independent Impostor Models

The optimal choice of anti-speaker or impostor model is not yet clear. There is evidence to support speaker independent models, cohorts and random anti-speakers. Speaker independent models are used in the experiments in Chapter 5.

De Veth (de Veth *et al.*, 1993) compared speaker independent models with groups of cohorts and found the SI models to be superior on one database (2.1%EER instead of 2.6%). On another database, however, when multiple recording sessions were used for the test data, SI models were not as good as using the N-best cohort scores. This result is therefore inconclusive.

Rosenberg (Rosenberg *et al.*, 1992) found that using the mean of the cohort scores was best. If the mean statistic is also best for random anti-speaker sets, then the argument for using speaker independent impostor models is strengthened. This is because using the average of a group of randomly selected speaker dependent models is similar to using a model trained using data from

[18] Again using same-microphone conditions.

a group of randomly selected speakers. If a speaker independent model can be shown to be as effective for speaker normalisation as a set of anti-speaker models, then it should be preferred since the computational load of using a speaker independent model is $\frac{1}{k}$ that of using a set of anti-speaker models. As stated previously, if the anti-speakers are randomly chosen then there is reason to believe that k should be made as large as possible, in order to get a robust average. If speaker dependent models are used, then k is limited by the linearly increasing computational load. In a speaker independent model the number of speakers used to train the model does not affect the computational load during verification, so a robust averaging of data from a very large number of speakers can be obtained without any increase in computation during verification.

Matsui (Matsui & Furui, 1994a) used an *a posteriori* probability approach to normalisation, rather than using the log-likelihood ratio. This involves summing the likelihood scores from all speaker models, including the client model. In order to avoid this computation the use of speaker independent impostor models, termed *pooled* models, was investigated. Matsui proposed two methods to construct the speaker independent impostor models. Method A involves training an SI model from enrolled speakers and adapting the model as new clients are enrolled on the system. Method B involves storing all the enrolment speech from all clients and re-training a speaker independent impostor model each time a new client is added. This requires additional storage space and a considerable amount of time re-training. A potential disadvantage with both these methods is that thresholds may be difficult to stabilise since the impostor model is frequently changing. On a text-prompted task Matsui found that Method A and Method B perform 30-50% better than using the *a posteriori* probability approach to normalisation. Other experiments showed that the *a posteriori* probability approach to normalisation produces very similar results to the log likelihood ratio approach (Furui, 1994), so it can perhaps be concluded by extrapolation that speaker independent models will be also be superior to a random anti-speaker set using log likelihood ratio normalisation.

The method used in HASAS for constructing a speaker independent impostor model is that of (Carey & Parris, 1992) which involves using a separate group of speakers to train a speaker independent impostor model. This does not involve any storing of speech or re-training, but it relies on being able to collect appropriate speech data to train SI models for all applications.

Note that the use of a speaker independent model as an impostor model has a EII bias problem if speakers from the impostor set are used to train the impostor model. The size of the

bias, however, is much less than for cohort speakers if the impostor model is trained using a large number of speakers. The speaker independent models used in HASAS in Chapter 5 use data from 80 different speakers so that the match between the impostor model and any particular impostor will not be great. The possibility of EII bias resulting from the use of impostor speakers to train the speaker independent models is investigated in Section 5.7.

Normalisation for Robustness to Microphone Variation

Speaker normalisation using cohort speakers has been shown to be particularly effective when there is a mismatch between the handset microphone used for training and that used for testing (Rosenberg *et al.*, 1992). In this experiment the client test data was recorded using an electret microphone but the training data was recorded using a carbon button microphone. The impostor test data was also recorded using a carbon button microphone, so there is no microphone mismatch between client model and impostor test data.

Since the client trials had a microphone mis-match and the impostor trials did not the SS EER increased dramatically from 2.9% to 22%[19]. By using cohort speaker models for normalisation the EER was reduced back down to 4.8%. The cohort models were also trained on carbon button microphones. This clearly shows the effectiveness of what Rosenberg describes as a *dynamic threshold*. The client test data has a mismatch with the client model and the likelihood score is therefore reduced, but it also has a mis-match with the cohort model so the normalising score which is subtracted is also reduced. The impostor test data has no microphone mismatch with the client model so it has a relatively high likelihood score, but it also has no mismatch with the cohort model so the cohort likelihood score is also high. While this result makes normalisation appear to be a promising technique for coping with channel or microphone mismatch, not all cases have been considered, as the following analysis will show.

In a real application there is unlikely to be any control over the microphone which is used when enrolling, so the client and cohort models could be trained using different microphones. The cohort selection procedure proposed in (Rosenberg *et al.*, 1992) encourages but does not ensure[20] a microphone match between client and cohort models. We make the following definitions.

[19] Using an impostor set with the proposed cohort speakers excluded
[20] Assuming the potential cohort models have a mix of microphone types

λ_{C1}	Client model trained using a carbon button microphone
λ_{C2}	Client model trained using an electret microphone
λ_{I1}	Impostor model trained using a carbon button microphone
λ_{I2}	Impostor model trained using an electret microphone
X_{C1}	Client test utterance recorded using a carbon button microphone
X_{C2}	Client test utterance recorded using an electret microphone
X_{I1}	Impostor test utterance recorded using a carbon button microphone
X_{I2}	Impostor test utterance recorded using an electret microphone
H	Likelihood score from a match between a client model and client data or an impostor model with impostor data. H stands for high.
L	Likelihood score from a match between a client model and impostor data or an impostor model with client data. L stands for low.
$+\Delta$	Change in likelihood score when training and test microphones match
$-\Delta$	Change in likelihood score when training and test microphones do not match

Leaving aside microphone variation for a moment the traditional verification decision is described in Table 2.1. The effect of speaker normalisation is given in Table 2.2.

Test Data	Verification Score	Decision
X_C	H	Accept
X_I	L	Reject

Table 2.1: The traditional verification decision. Making the correct decision depends on H being consistently greater than L.

Now consider the microphone variation. There are four modelling possibilities, and these

Test Data	Normalised Score	Decision
X_C	$H - L$	Accept
X_I	$L - H$	Reject

Table 2.2: The effect of speaker normalisation. Making the correct decision depends on H-L being consistently greater than L-H.

can be split into two groups.

$$\left.\begin{array}{ll} \lambda_{C1} & \lambda_{I1} \\ \lambda_{C2} & \lambda_{I2} \end{array}\right\} \text{No mismatch between models}$$

$$\left.\begin{array}{ll} \lambda_{C1} & \lambda_{I2} \\ \lambda_{C2} & \lambda_{I1} \end{array}\right\} \text{Mismatch between models} \tag{2.6}$$

Test Data	Normalised Score	Decision
X_{C1}	$(H + \Delta) - (L + \Delta) = H - L$	Accept
X_{C2}	$(H - \Delta) - (L - \Delta) = H - L$	Accept
X_{I1}	$(L + \Delta) - (H + \Delta) = L - H$	Reject
X_{I2}	$(L - \Delta) - (H - \Delta) = L - H$	Reject

Table 2.3: Normalisation scores for the case of (λ_{C1},λ_{I1}).

Test Data	Normalised Score	Decision
X_{C1}	$(H + \Delta) - (L - \Delta) = (H - L) + 2\Delta$	Accept
X_{C2}	$(H - \Delta) - (L + \Delta) = (H - L) - 2\Delta$	?
X_{I1}	$(L + \Delta) - (H - \Delta) = (L - H) + 2\Delta$	?
X_{I2}	$(L - \Delta) - (H + \Delta) = (L - H) - 2\Delta$	Reject

Table 2.4: Normalisation scores for the case of (λ_{C1},λ_{I2}).

The normalisation score for all combinations of models and test utterances for the (λ_{C1},λ_{I1}) model combination is given in Table 2.3. In all cases normalisation works well, and the very successful experiments using cohort normalisation in (Rosenberg *et al.*, 1992) which were described earlier deal with this case. Clearly the other case where there is no microphone

mismatch between models (λ_{C2},λ_{I2}) will also work well.

We will now look at the case where there is a microphone mismatch between the client and the impostor models. Table 2.4 gives results for the case of (λ_{C1},λ_{I2}). For X_{C1} and X_{I2} type test utterances the normalisation enhances the verification process, making a correct decision more likely. For X_{C2} and X_{I1}, however, the normalisation process makes the correct decision *less* likely, with the difference between client score for test utterance X_{C2} and impostor utterance X_{I1} being reduced by 4Δ compared to the un-normalised case.

Some estimate of how significant this is can be gained from the un-normalised cross-microphone experiment in (Rosenberg *et al.*, 1992). In this case we have the situation given in Table 2.5. The cross-microphone conditions reduce the difference between client and impostor

Test Data	Verification Score	Decision
X_C	$H - \Delta$	?
X_I	$L + \Delta$	?

Table 2.5: The un-normalised verification decision under Rosenberg's cross-microphone conditions. Making the correct decision depends on H being consistently 2Δ greater than L.

scores by 2Δ. This is sufficient to increase the EER from 2.9% to 22%.

Obviously cases where the difference between client and impostor scores is reduced 4Δ are likely to produce errors. If the microphone type was a random variable the 4Δ degradation would occur 25% of the time. Fortunately the microphone type used for enrolment and for verification will not be random in most applications. It is likely that the client will usually call from the same phone that they used for enrolment. If a client has different microphone types at home and at the office, say, then the client microphone type may be more or less random. The impostor microphone type will be distributed according to the proportion of the two microphone types in the general population.

The best way of reducing the change of the 4Δ degradation is to avoid a microphone mismatch between client and cohort models. If the system designer ensures that there is a good range of cohort speakers trained on all microphone types, the cohort selection process will tend to favour a match in microphone type between client and cohort model. The same approach can be used if a speaker independent model is used for normalisation.

2.4.5 Discussion of Recent ASV Systems

In (Rosenberg *et al.*, 1990b) a sub-word based HMM system was evaluated on an isolated digit database recorded over dialled up telephone lines. The database was similar in size and design to that used to evaluate HASAS. In Rosenberg's database, however, the speakers were in a sound booth and there was no handset variation.

Rosenberg's choice of HMMs over template based techniques was based on the [then] recent success of HMMs in large vocabulary ASR systems (Rabiner *et al.*, 1989; Lee *et al.*, 1990).

The main difference between Rosenberg's system and HASAS is that sub-word CHMMs are used instead of whole-word SCHMMs with state duration modelling. The models are trained using 8 tokens of each digit (compared to just 5 in HASAS) and these tokens are taken from 2 training sessions so some temporal variation is included in the training data. HASAS training data comes mainly from a single session, however the nature of the database means that the 5 utterances occasionally come from more than one session. All other factors being equal , the difference in the amount of training data, the presence of background noise and the handset variation in the HASAS database should mean that HASAS will not perform as well as Rosenberg's system.

Some indication of the effect of varying the amount of training data can be gained from Rosenberg's experiment in which increasing the amount of training data from 8 to 10 utterances reduced the speaker specific (SS) EER from 1.8% to 1.1%. This is not just a result of the 25% increase in the amount of training data but also because data from 3 rather than 2 recording sessions are used for training[21]. This not only improves the modelling of temporal (intra-speaker) variation, but also means that 2 of the test utterances are from the same recording session as two of the training utterances.

HASAS experiments, detailed in Table B.17 for a 7 digit-sequence, using 5 training tokens per digit produced an SS EER of 2.84%. This compares favourably with the 1.8% SS EER of Rosenberg's system, considering the differences in the databases in terms of the amount of training data, background noise, and handset variation. Note also that the performance of Rosenberg's system levels off after 7 digits, whereas the performance of HASAS continues to drop to 1.4% using 12 digits.

[21] Four utterances of each digit were recoded in each session.

Later work by Rosenberg (Rosenberg *et al.*, 1991) used an average of 4.4 tokens/ digit of training data, which is more directly comparable with HASAS, but the database was of connected digits. The SS EER using a 12 digit test utterance was 1.7% compared to 1.4% for HASAS. It would be expected that HASAS would perform better, since isolated digits are used instead of connected digits, although HASAS did have to cope with background noise and handset variation.

Unfortunately, these are the closest comparisons with other ASV systems that are possible. Absolute performance is so strongly dependent on the exact task, and the database used, that most comparisons mean little. The systems, databases and results of Rosenberg's system and HASAS are sufficiently similar to conclude, however, that the HASAS system is representative of HMM-based ASV systems. It is reasonable to conclude, then, that algorithms which improve the performance of HASAS can be expected to improve other HMM-based ASV systems. It is on this basis that the research presented here constitutes a useful contribution to the ASV field.

Chapter 3

The HASAS system

3.1 Introduction

This chapter is concerned with describing the SCHMM ASV system, known as HASAS (HMM Automatic Speaker Authentication System), which was used in these experiments, and explaining some of the design constraints which were imposed.

The chapter begins with a description of the task for which this system would be used, were it to be adopted commercially. A set of design constraints is then established which will enable the HASAS system to achieve the prescribed task. These constraints come from the following three design choices:-

- The modelling techniques used, namely hidden Markov models.
- The database which was available.
- The research goals of this thesis.

Section 3.3 provides a description of the database that was available to assess the performance of HASAS, together with details of the way in which it was used and the ways it has influenced the specification of HASAS.

Section 3.4 gives a full specification of HASAS, which was designed and implemented by the author in order to achieve the research goals of this thesis. The section begins with a description of SCHMM, as used in HASAS, to a level of detail sufficient to enable a reader familiar with HMM theory to implement HASAS themselves. Following the description of the model architecture, there are details of the other elements that make up an ASV system, such as feature extraction techniques, state segmentation, the calculation of the verification score, and

the decision logic applied to the score in order to make the accept/reject decision.

Although the various elements of HASAS such as SCHMM, Gaussian state duration modelling and multiple codebooks are well known in the literature, the combination used in this thesis has not been used for ASV before. The most novel feature of the architecture is the common state segmentation using cepstral features which runs through both training and testing of the multiple codebook models. This approach comes from a view of ASV being a combined speech and speaker modelling process and this point of view has led to many of the successful new techniques described in this thesis.

3.2 System Design Objectives

3.2.1 Task Definition

Telephone banking and telephone credit card authorisation are the applications which were the focus of this research. The scenario is one of an automatic speech recognition (ASR) system working in tandem with an ASV system to provide automatic verification of an account number or credit card number. The client should preferably be able to speak using connected speech rather than isolated digits. A text-dependent system is appropriate to this task.

Each client would be required to enrol with the bank or credit card company before using the system. This enrolment must be as fast and convenient as possible, and must ideally be done in a single call. The system must work over standard dialled-up telephone lines and be robust to variations in the client's calling location and telephone handset specification.

3.2.2 Design Constraints

Computational Requirements

The computation requirements must be such that real-time operation is realistic. A fast, modern micro-processor or digital signal processor can be assumed to be available.

Four standard speech recognition feature sets have been used in this thesis (LPC cepstra, MFCC, and their difference coefficients). If the same features are used for ASV and ASR in a given application there is a computational bonus that the feature extraction can be shared by the ASR and ASV systems.

Model Storage Requirements

While the storage requirements of the client model must be kept as low as possible this constraint is not acute. For a telephone banking task it can be assumed that the client's models do not need to be stored on a magnetic or smart card, so the size of the models is not critical.

It can also be assumed that although the number of clients is likely to be very large in some cases, the client model data will be stored centrally and so the overall amount of storage space available for client models can be in the range of several gigabytes. A reasonable assumption is that anything less than about 10kbytes per client is acceptable.

Training Data

The client models in an ASV system are necessarily speaker dependent. This means that data must be collected from each client before they can use the system. The amount of data collected and the way in which it is collected will strongly affect the amount of inconvenience a customer must tolerate in order to use the system. This will relate directly to customer acceptance of the technology. For this reason the amount of training data will always be constrained in some way. The severity of the constraint will depend on the application. If the ASV is for the door entry system of a company, where security requirements are high and clients are very co-operative, it would be reasonable to request significant quantity and quality of training data. Three or four training sessions, taking 10 to 15 minutes each, would not be unreasonable.

HASAS on the other hand, is intended for telephone applications to be used by thousands of clients from the general public. The amount of time and inconvenience involved in enrolling a client could well be the second most significant factor (after false rejection rate) in a client deciding whether to use the system.

Section 2.4.2 explained why model adaptation will be vital to the long-term performance of any ASV system, using additional data to both improve the models and adapt to gradual changes in the clients speech. The initial performance, however, must be satisfactory, and this must be done using only the training data obtained at enrolment.

Ideally, several sessions over a period of time should be used for enrolment but this is probably unrealistic for telephone applications because it would be inconvenient for the clients to have to enrol several times. The clients are, after all, enrolling in the system because they

want to save on time and inconvenience.

HASAS was therefore designed to achieve the highest possible performance from a training set consisting of 5 repetitions of each digit. Note that the nature of the available database meant that the 5 repetitions do not necessarily come from the same recording session.

Isolated Digits

Although it was considered that connected digits represent a more realistic match to potential applications than isolated digits, the limitations of the available database meant that isolated digits were used instead of connected digits, since only isolated digits were available in sufficient numbers.

It is difficult to extrapolate the performance of an isolated word ASV system to connected digit performance but absolute performance is not the measure of interest in this thesis. What is more important is to know whether the performance improvements obtained on the isolated digit database used in this work will correspond to similar improvements on a connected digit task. This can only be determined by further practical experimentation on a connected digit database.

Word or Sub-Word Models

In general, the larger the acoustic unit being modelled, the more reliable the model will be. It is therefore desirable to use the largest unit possible. Since the task definition requires the use of variable length sequences of digits, it is not feasible to model entire digit sequences. This is simply because there are a large number of possible sequences, and an excessive amount of training data would be required.

When it is required to have a flexible or expandable vocabulary, it is necessary to use sub-word unit models. From these models, words can be constructed which are not necessarily present in the training data. Since the task definition for this work requires a limited vocabulary consisting only of the digits, sub-word unit models are not necessary and whole-word models are used.

While HASAS was designed with whole-word models in mind, the the use of sub-word models was a consideration throughout. The result is that HASAS could also be used with sub-word models by simply phonetically labelling the data and retraining using sub-word models.

The choice of sub-word units would, however, have to be carefully considered in the light of the severe constraint imposed on training data by the task definition, in particular the issue of whether they should be context dependent or context independent.

3.2.3 Form of HMM models

Continuous HMMs (CHMM) and SCHMMs are now widely preferred over discrete HMMs (DHMM) in the literature because continuous probability densities offer more robust coverage of feature space than do discrete vector quantisation codebooks of a finite size (Huang & Jack, 1988; Forsyth & Jack, 1993).

The decision to employ SCHMM rather than CHMM was based on the strict limitations on the amount of training data available.

It is intuitive, and supported in the literature (Rabiner & Hwang, 1992) (Section 6.8) that semi-continuous is more effective than full continuous models when there is little training data. This is because in continuous HMM the means and variances of the probability density functions must be estimated for each state of the model. In SCHMM only a weight for each mixture must be estimated.

With a 6 state CHMM model, using 4 mixtures per state, and 5 training tokens with an average length of 80 frames, each mixture is strongly influenced by only $(80\times5)/(6\times4) = 16$ vectors. From this a mean and variance must be calculated.

In a SCHMM system with a codebook of size 32 a weight must be estimated for each codeword probability density function using all $(80\times5)/6 = 67$ vectors.

3.2.4 Summary of Design Constraints

The following design constraints were applied to the system.

- Isolated digits.
- Whole word models.
- Telephone speech.
- 5 training tokens per digit.
- Real-time operation must be realistic.

- Model storage requirements must be kept under 10kbytes per client.

3.3 Database

A subset of British Telecom's[1] *BRENT* speaker verification database was used for these experiments. The data was recorded over dialled-up telephone lines in the United Kingdom[2].

The database contains isolated digits from 120 male and female speakers. The subjects are native British English speakers from throughout the United Kingdom. The database was collected with pauses between digits, so that there were no co-articulation effects between digits. Around fifty repetitions of each digit were collected from each of the speakers.

3.3.1 Handsets

It is desirable that HASAS should be capable of coping with a client using a variety of telephone handsets and, therefore a variety of microphone types. As discussed in Section 2.4.4 the database should have variation in handsets which is realistic to the task. It proved difficult to enforce microphone variation explicitly, but the subjects supplying the speech data were encouraged to use a variety of handsets.[3] Although this makes it difficult to make comparisons as to difficulty of the BRENT database relative to databases where the microphone type is precisely known, it does mean that the database contains a realistic variation in microphone types.

3.3.2 Quality Control

Each utterance was graded by a human listener using a three category classification system.

- BAD: All utterances which did not contain the correct speech, or were not from the correct speaker. All utterances containing noise or distortion such that the digit could not be recognised by a human listener.
- O.K: All utterances not classified as BAD but containing significant quantities of distortion or noise, breathiness or lip-smack (clicks generated by the lips separating).
- GOOD:All other utterances

[1] Thanks to BT for the use of this database

[2] Some of the database specifications are proprietary and cannot be reported here.

[3] One subject even called from a telephone in his local bar!

Only utterances labelled *GOOD* were used in this study. This is justified using the assumption that the conditions required for a GOOD utterance are a reasonable standard of speech production and line conditions to demand for the use of an ASV system. The amount of noise or distortion required to be classified as O.K. is at a level such that most users would describe the connection as a *bad line*, or the speech as *noisy*. It is assumed that users will accept increased false rejection under such conditions.

The requirement that the user avoid excessive lip-smack and breathiness before or after the utterance is more difficult to justify as realistic, since some subjects were consistently unable to achieve GOOD utterances. However, no explicit instruction to avoid breathiness or lip-smack was given, and it is possible that doing so could greatly reduce the severity of these factors.

Much attention is paid to making databases appropriate to the *real world*, and this is sensible. It should be noted, however, that significant modification of user behaviour is possible with new technologies. Technologies such as automatic teller machines and microwave ovens place what were initially quite difficult restrictions on a user's behaviour, but these restrictions are now readily met by the majority of users, as they have learned what is required in order to get the most out of the technology. The same process would apply to automatic telephone banking. Users will quickly appreciate that it is a very useful service but they have to speak clearly and carefully, on a clear line, and from a quiet environment in order to get the best service.

3.3.3 Client Speakers

A subset of 21 speakers, 11 male and 10 female, who had 25 or more GOOD utterances for each digit were chosen as the set of client speakers. These speakers were enrolled in the system and models of their speech were created. The 25 utterances of each isolated digit were divided into 5 blocks (labelled A to E) each of 5 tokens. Each block was used to train an isolated digit model, creating 5 models of each digit for each client speaker.

If the A block data is used for training, then the B to E blocks can be used for testing. This process means that for each model there are 20 test utterances. This procedure is often referred to as jack-knifing.

The 25 utterances come from a series of sessions over the space of 6 months. The 25 utterances are in chronological order, but there is no guarantee that the 5 utterances in a given block come from the same training session or that they come from different sessions to the

utterances in the blocks on either side.

3.3.4 Codebook Speaker Set

A standard set of codebooks is used for all speakers and for all the digits. These codebooks have to be created from speech data typical to the task but independent of the client or impostor speaker data.

The reason the data used to train the codebooks should be from an independent set of speakers is that this represents the likely operating conditions for a telephone banking task. The system will need to be set up, and the codebooks created, before the client speakers are known. Impostor speakers are never known.

A group of 19 speakers (9 male and 10 female) form the *codebook speaker set*. The codebook for each feature set is created using one utterance of each digit from each speaker.

3.3.5 Impostor Speaker Set

The impostor set consists of 80 speakers who are not in the client or the codebook speaker sets. In addition to this, all client speakers other than the client speaker being tested are used as impostor speakers, making a total of 100 impostor speakers. Only one utterance of each digit is used from each impostor speaker making a total of 500 impostor utterances for each digit for each speaker after jack-knifing.

The partitioning of the database into client, codebook and impostor sets is illustrated in Figure 3.1.

3.3.6 Silence Removal

All utterances had excess silence removed in order to conserve storage space and reduce processing time. The algorithm used was very weakly constrained so as to avoid any errors. A margin of 200 ms from the cut-off suggested by the end-point detector was allowed at either end to further reduce the chance of error.

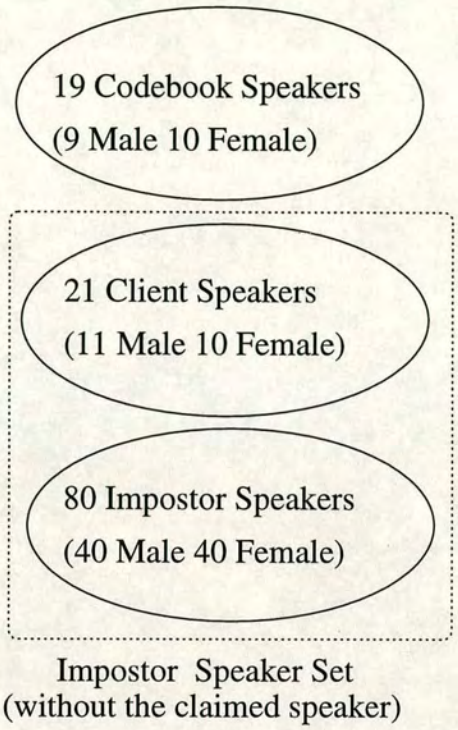

Figure 3.1: Partitioning of the database into client, codebook and impostor sets.

3.4 HASAS Specification

The following section describes the details of the HMMs used in the HASAS system. The models are actually Hidden Semi-Markov models because the duration modelling does not strictly obey the Markov independence assumption. The abbreviation to HMM is still used here for convenience.

3.4.1 Notation

A summary of mathematical notation is given in Table 3.1.

The main features of the architecture are as follows.

1. Strict left-to-right models with no skipped states.

   $a_{i,j} = 0, j \neq i+1,\ a_{i,j} = 1, j = i+1$

2. 6 state word models.

3. Gaussian state duration modelling is used throughout.

4. 32 probability density function codewords for each feature set.

Symbol	Definition
N	Total number of states
$S = s_1, s_2 \ldots s_N$	Set of possible states
$Q = q_1, q_2 \ldots q_T$	Optimal (Viterbi) state sequence
T	Total number of frames of speech in the utterance
t	time (frame number)
M	number of observation symbols (32 in this thesis)
m	codeword index
$\mu_{m,i}$	the i^{th} dimension of the mean of the m^{th} codeword
$\sigma^2_{m,i}$	the i^{th} dim of the diagonal co-variance of the m^{th} codeword
K	number of training utterances
k	training utterance index
$x(t)$	Feature vector - e.g. LPC cepstral coefficients.
$x_i(t)$	The i^{th} dimension of $x(t)$
$V = \{v_1, v_2 \ldots v_M\}$	Set of codewords
$C_i = \{c_{i,1}, c_{i,2} \ldots c_{i,M}\}$	Set of codeword weights for state s_i
$a_{i,j}$	Transition prob from s_i to s_j.
$b_i(t) = \sum_{m=1}^{M} c_{i,m} \times P(x(t)\|m)$	observation probability of state s_i for frame t.
$\lambda = (A, B, D, \pi)$	HMM
$O = O_1, O_2 \ldots O_T$	Observations
B	Observation Probabilities
A	Transition Probabilities
$D = \{d_1, d_2 \ldots d_{\tau_{max}}\}$	State duration probabilities
τ	State duration index.
τ_{max}	Maximum state duration (150 frames)

Table 3.1: Table of symbols.

3.4.2 Feature Extraction

The feature sets used throughout Chapters 4 and 5 are LPC based cepstral coefficients, mel-frequency cepstra coefficients (MFCC) and their first order differentials, referred to as Δcepstra and ΔMFCC respectively.

The signal processing is standard and can be found in (Rabiner & Hwang, 1992)[4]. A first order pre-emphasis filter of the form

$H(z) = 1 - 0.97z^{-1}$

was applied to the speech waveform to spectrally flatten the signal. A Hamming window of 20ms was used with a 15ms shift between frames. The standard technique for LPC cepstral

[4]The HCode software from the HTK-4.1 software package was used for all feature extraction.

calculation via autocorrelation co-efficients and LPC parameters was used, as described in Section 3.3.7 of (Rabiner & Hwang, 1992). The 12 cepstral coefficients are calculated via 15^{th} order LPC analysis, and a bandpass lifter with de-emphasis point at the 15^{th} coefficient. The mel-frequency cepstral co-efficients were calculated via fast Fourier transform (FFT). As pointed out in (Openshaw *et al.*, 1993) the LPC calculation produces a smoother spectrum and it is possible that using the LPC and the FFT based approaches to cepstral analysis may increase the independence between the LPC cepstra and MFCC feature sets. The difference coefficients are taken across a window of +/-2 frames for both cepstra and MFCC features.

3.4.3 Codebooks

In HASAS the codebooks used are common to all states of all the models. A codebook consists of M Gaussian probability density functions (PDF), whose mean and diagonal covariance vectors are estimated using the standard k-means clustering algorithm on 30,000 vectors from the 20 codebook speakers. Four separate codebooks were constructed, one for each feature set.

In applying the codebook to a feature vector $x(t)$ (for instance an LPC cepstral vector) the probability $P(x(t)|m)$ is calculated for each codeword m using Equation 3.1.

$$P(x(t)|m) = \sqrt[n]{\prod_{i=1}^{n} \left[\frac{1}{\sigma_{m,i}\sqrt{2\pi}} . e^{-\frac{1}{2}\left(\frac{x_i - \mu_{m,i}}{\sigma_{m,i}}\right)^2} \right]} \qquad (3.1)$$

Where x_i is the i^{th} dimension of the feature vector $x(t)$, $\mu_{m,i}$ is the i^{th} dimension of the mean of the m^{th} codeword and $\sigma^2_{m,i}$ is the i^{th} dimension of the diagonal covariance of the m^{th} codeword.

The representation of the t^{th} frame of speech has been transformed from the n dimensional cepstral vector $x(t)$ to an M dimensional vector of probabilities.

Word or Speaker Specific Codebooks

The codeword means and variances were not re-estimated during training to make speaker or word specific codebooks. Although it is likely that re-estimation of the means at least would improve the speaker dependent model, codebook re-estimation was initially rejected in order to keep the models simple. In the light of the work in Chapter 5 it is not clear whether codebook re-estimation would be beneficial. It is possible that having a standard codebook for all speakers may improve the robustness of comparisons between models.

Separate codebooks for the different digits would be likely to be beneficial to the speaker verification stage of HASAS since it would improve the quality of the models. Having digit specific codebooks might have implementation difficulties if connected digits were to be used in the system, since the recognition stage would have to segment the utterances into words before the codebooks could be used. This would entail a break from the approach of keeping as much commonality as possible between the speech recognition and speaker verification stages. The speech recognition stage and the speaker recognition stage would have to use different codebooks. If the improvement in performance is significant then this inconvenience would be acceptable. The use of speaker or word specific codebooks would be a useful subject of further experimentation.

3.4.4 Number of States

The weights of each state determine the area of the feature space that the state models. If an acoustic event is defined as a period during which the speech signal is confined to a particular region in feature space then the number of states in a model should correspond roughly to the number of acoustic events in the utterance being modelled. Experimentation can be done to determine the optimal number of states for each digit, as was done for the ASR system described in (Buhrke *et al.*, 1994). All HASAS word models consist of six states, which preliminary experimentation showed to be a reasonable figure.

3.4.5 State Duration Modelling

Standard HMMs have a set of transition probabilities a_{ij} to represent the probability of a transition from $s_i \rightarrow s_j$. This means that the probability of remaining in s_i for exactly τ frames $d_i(\tau)$ is given by.

$$d_i(\tau) = a_{ii}^{\tau}(1 - a_{ii}) \tag{3.2}$$

A plot of $d_i(\tau)$ against state duration τ in Figure 3.2 shows that the fixed transition probabilities produce an exponential state duration model. This is very poor model of state duration and it is a result of the Markov assumption that q_t depends only on q_{t-1}. If we relax this constraint and allow q_t to depend on $q_{t-1}, q_{t-2} \ldots q_{t-\tau_{max}}$, where τ_{max} is the maximum state

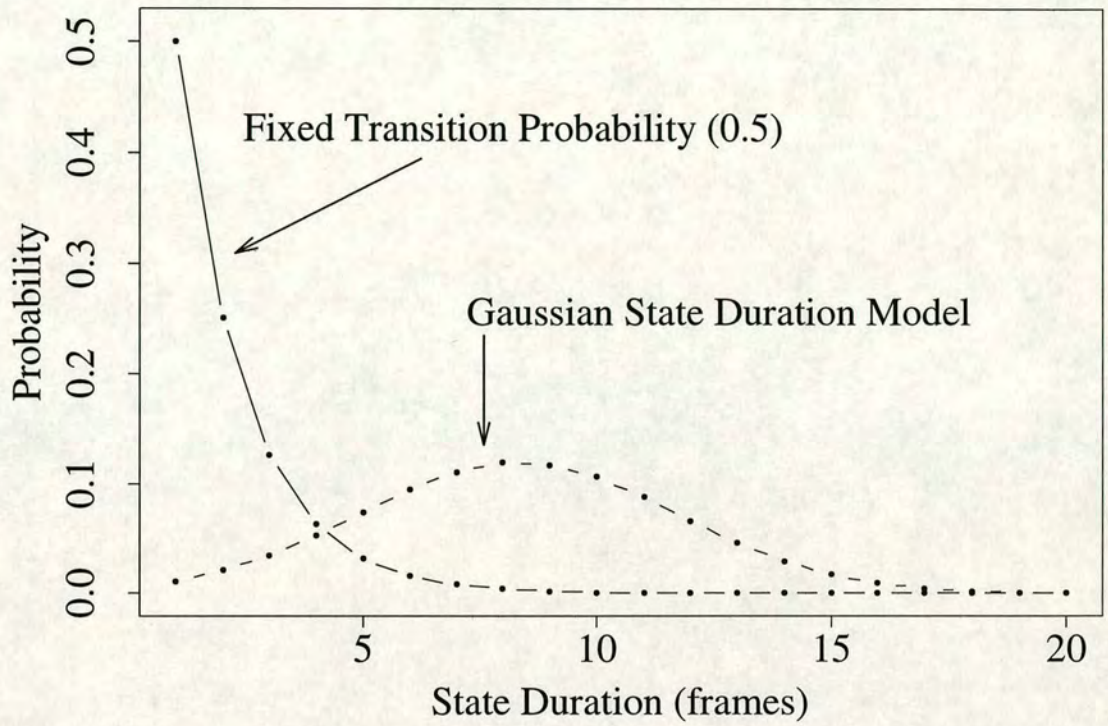

Figure 3.2: State duration probabilities with fixed transition probabilities compared to that obtained using explicit Gaussian state duration modelling.

duration, we can explicitly model the state duration. Several different parametric forms of state duration model have been proposed. Gaussian state duration models were used in HASAS and the resulting state duration probability distribution is given in Equation 3.3 where $d_i(\tau)$ is the probability of being in state s_i for exactly τ frames.

$$d_i(\tau) = \frac{1}{\sigma_i\sqrt{2\pi}}.e^{-\frac{1}{2}\left(\frac{\tau-\mu_i}{\sigma_i}\right)^2} \tag{3.3}$$

The duration probabilities $d_i(\tau)$ are normalised according to Equation 3.4.

$$\sum_{\tau=1}^{\tau_{max}} d_i(\tau) = 1 \tag{3.4}$$

The maximum allowed duration τ_{max} has a strong effect on the amount of computation in the Viterbi algorithm ($O(\tau_{max}^2)$). In theory τ_{max} should be made slightly more than the maximum state duration that will occur. In this experimental system it was sufficient to make it *comfortably high enough*, since computational efficiency was not critical. The maximum duration was set to $\tau_{max} = 150$ throughout this work.

Note that the models are now semi-Markov models since the Markov constraint has been relaxed. Explicit state duration modelling provides a more accurate description of state duration probabilities than is possible with fixed transition probabilities - there is a mean representing the most likely state duration and a variance around this. Using conventional transition probabilities the most likely state duration is always one frame, which does not reflect the true nature of the speech signal.

State duration modelling has been shown to be successful in speech recognition (Levinson, 1986), although the increase in computation required often prevents it being used. The increase in computation can easily be accommodated by HASAS, however, since knowledge of the text of the utterance means that the search space of the Viterbi algorithm is small, compared to ASR.

3.4.6 Seeding the Models

The weights and state duration probabilities of the models must be initialised in some way before they are re-estimated during training. Firstly a speaker independent model of each digit was constructed from the codebook speaker data. These models were initialised with equal weights for all the codewords and flat duration models (i.e the probabilities of all durations were set to $1/\tau_{max}$).

Once these speaker independent models were established they were used to initialise or *seed* the training of another set of speaker independent models, which were trained using one token from each of the 80 non-client speakers. These 80-speaker speaker independent models were then used to seed the training of the client models[5], thereby ensuring a similar correspondence between states and acoustic events in all models.

The implications of the lack of independence in training and test data are not significant. The impostor data provides a model which provides a starting point for the training of the speaker dependent models. The relationship of the speaker dependent models to the impostor data will be very slight and will result in increased false acceptance rate if it has any effect. The 80-speaker speaker independent models are also used as reference models in the techniques described in Chapter 5. The implications of this are discussed in Section 5.7.

[5] Seeding speaker dependent models using speaker independent models was proposed in (Rosenberg *et al.*, 1991), and shown to be particularly effective when training data is limited (average of 4.4 tokens/digit).

3.4.7 Silence Model

A model of silence is needed for segmentation. A single state silence model was trained using an equal weight seed model with state duration probabilities of $1/\tau_{max}$. Silence data was clipped from the front and back of a few of the codebook speaker utterances. The codeword weights were re-estimated using Baum-Welch while the state duration probabilities were not re-estimated and remained equal to $1/\tau_{max}$ because it is desirable for the duration probability to be constant, regardless of the length of the silence. The silence model was used before and after each digit in a silence-digit-silence configuration. Many systems also use noise models to model such things as line clicks, and lip-smack. Noise models would undoubtedly help in obtaining a good state segmentation but they are not used in these experiments.

3.4.8 Training

All speaker dependent client models were seeded from the eighty-speaker models and trained using five occurrences of a single isolated digit. The weights for the codewords were re-estimated using the Baum-Welch algorithm. The algorithm used was derived by combining the algorithms given in (Huang *et al.*, 1990) for DHMM with Gaussian state duration modelling with those for SCHMM and is described below.

The use of a strict left to right model means that $q_1 = 1$ and $q_T = N$, which can be expressed in terms of initial state probabilities π as

$$\pi_i = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1 \end{cases} \tag{3.5}$$

The forward variable α is calculated recursively from the following.

$$\alpha_1(1) = d_1(1) \times b_1(1), i = 1$$

$$\alpha_i(1) = 0, i \neq 1$$

$$\alpha_1(t) = d_1(t) \times \prod_{l=1}^{t} b_1(l)$$

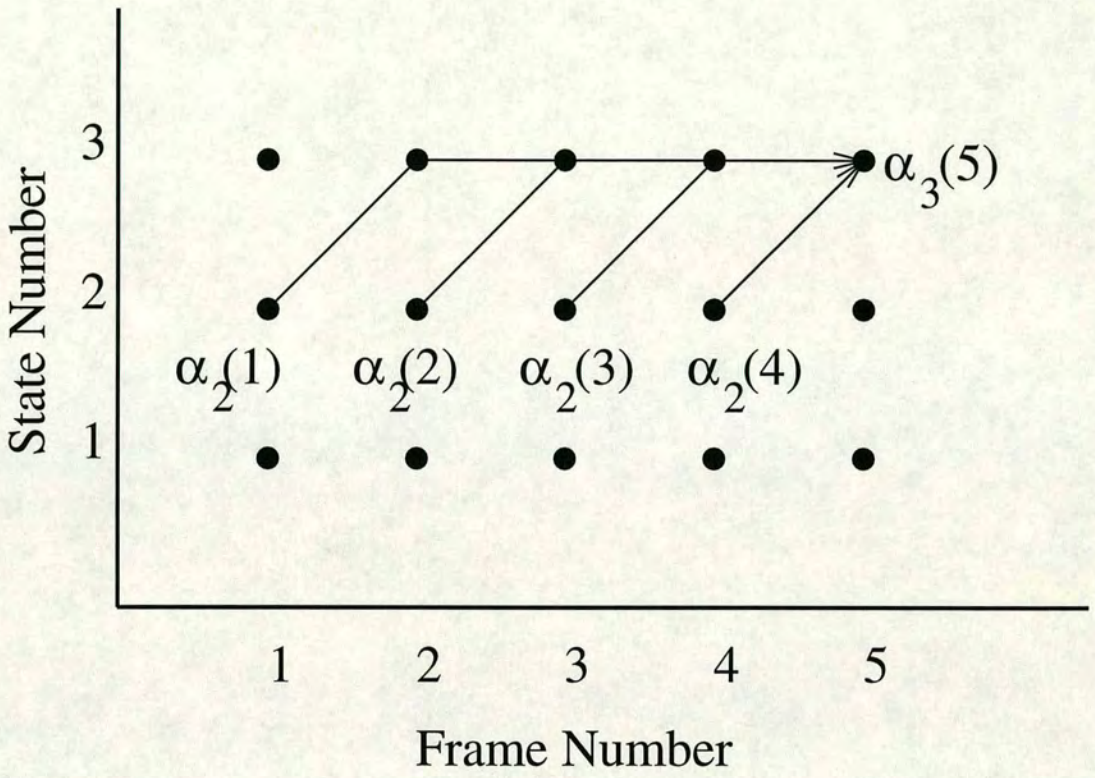

Figure 3.3: Illustration of the calculation of the forward variable $\alpha_3(5)$.

$$\alpha_i(t) = \sum_{\tau=1}^{\min(t-1,\tau_{max})} \alpha_{i-1}(t-\tau) \times d_i(\tau) \times \prod_{l=1}^{\tau} b_i(t-\tau+l), (i > 1) \tag{3.6}$$

Figure 3.3 gives an example of the calculation of the forward variable $\alpha_3(5)$.

$$\begin{aligned} \alpha_3(5) = \sum_{\tau=1}^{4} \alpha_2(5-\tau) \times d_3(\tau) \times \prod_{l=1}^{\tau} b_i(5-\tau+l) \\ = \alpha_2(4) \times d_3(1) \times b_3(5) \\ + \alpha_2(3) \times d_3(2) \times b_3(4) \times b_3(5) \\ + \alpha_2(2) \times d_3(3) \times b_3(3) \times b_3(4) \times b_3(5) \\ + \alpha_2(1) \times d_3(4) \times b_3(2) \times b_3(3) \times b_3(4) \times b_3(5) \end{aligned} \tag{3.7}$$

Likewise, the backward variable β is calculated recursively from the following.

$$\beta_N(t) = 0, (t < T)$$

$$\beta_N(T) = 1$$

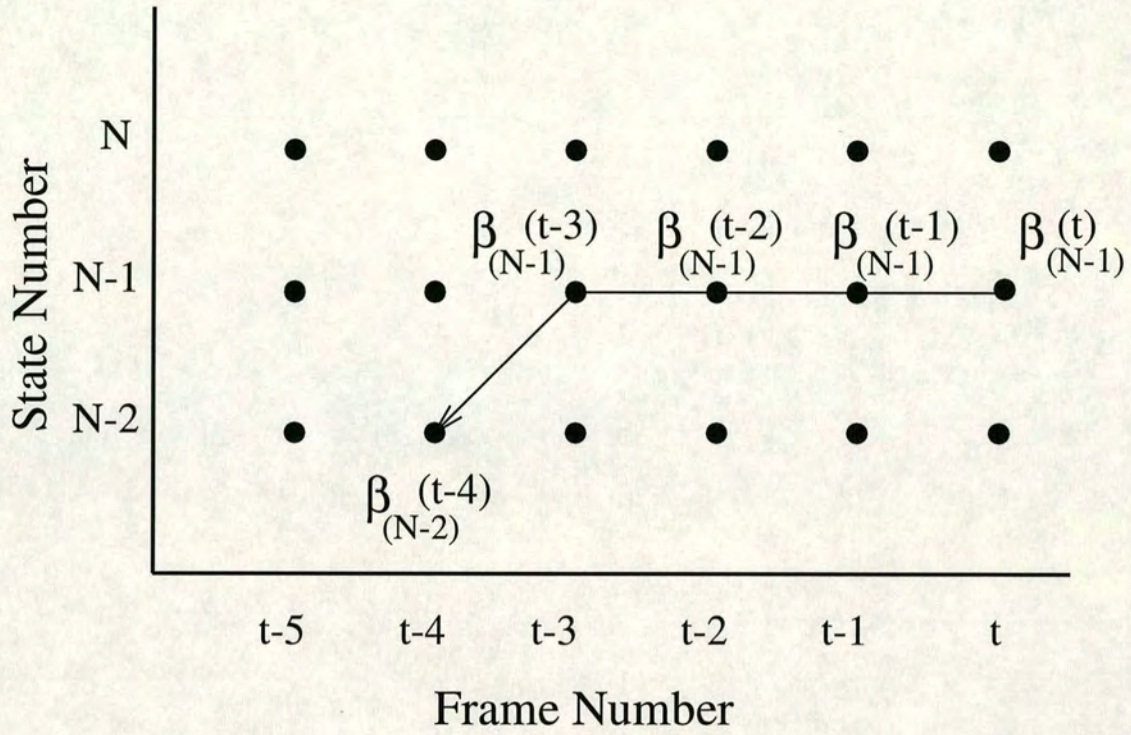

Figure 3.4: Illustration of the calculation of the backward variable $\beta_{N-2}(t-4)$.

$$\beta_i(T) = 0, i \neq N.$$

$$\beta_i(t) = \sum_{\tau=1}^{\min(T-t,\tau_{max})} \beta_{i+1}(t+\tau) \times d_{i+1}(\tau) \times \prod_{l=1}^{\tau} b_{i+1}(t+l) \tag{3.8}$$

Figure 3.4 gives an example of the calculation of the backward variable $\beta_{N-2}(t-4)$.

$$\begin{aligned}\beta_{N-2}(t-4) = \sum_{\tau=1}^{\min(4,\tau_{max})} \beta_{N-1}(t-4+\tau) \times d_{N-1}(\tau) \times \prod_{l=1}^{\tau} b_{N-1}(t-4+l) \qquad (3.9)\\ = \beta_{N-1}(t-3) \times d_{N-1}(1) \times b_{N-1}(t-3)\\ +\beta_{N-1}(t-2) \times d_{N-1}(2) \times b_{N-1}(t-3) \times b_{N-1}(t-2)\\ +\beta_{N-1}(t-1) \times d_{N-1}(3) \times b_{N-1}(t-3) \times b_{N-1}(t-2) \times b_{N-1}(t-1)\\ +\beta_{N-1}(t) \times d_{N-1}(4) \times b_{N-1}(t-2) \times b_{N-1}(t-1) \times b_{N-1}(t)\end{aligned}$$

The probability of being in state s_i during the interval [t:t+τ] and leaving at time $t+\tau$ is denoted $\gamma_{i,t,\tau}$.

$$\gamma_{1,t,\tau} = \frac{\alpha_{i-1}(t) \times d_i(\tau) \times \beta_i(t+\tau) \times \prod_{l=1}^{\tau} b_i(t+l)}{\alpha_T(N)} \tag{3.10}$$

If the superscript k denotes the kth training utterance in a set of K utterances, the codeword weight for the m^{th} codeword in state s_i ($c_{i,m}$) is re-estimated using Equation 3.11.

$$c_{i,m} = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{\sum_{t=1}^{T} \sum_{\tau=1}^{\min(T-t,\tau_{max})} \left[\gamma_{i,t,\tau}^{k} \times \prod_{l=t+1}^{t+\tau} P(v_m / x^k(l)) \right]}{\sum_{t=1}^{T} \sum_{\tau=1}^{\min(T-t,\tau_{max})} \gamma_{i,t,\tau}^{k} \times \tau} \right] \tag{3.11}$$

The numerator is proportional to the probability of occurrence of the codeword v_m in state s_i. The denominator is proportional to the probability of being in state s_i.

The duration probabilities $d_i(\tau)$ can be re-estimated using Equation 3.12.

$$d_i(\tau) = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{\sum_{t=1}^{T} \gamma_{i,t,\tau}^{k}}{\sum_{t=1}^{T} \sum_{l=1}^{\min(T-t,\tau_{max})} \gamma_{i,t,l}^{k}} \right] \tag{3.12}$$

The vector of duration probabilities is approximated by a Gaussian distribution, characterised by a mean and variance.

Although SCHMMs allow for the possibility of re-estimating the means and variances of the codebook probability density functions during training (Huang *et al.*, 1990), this was not done.

Common Segmentation Framework for Multiple Codebook Training

Several different feature sets were trained for each model, because it was not initially known which features would perform best and because it was hoped that a combination of feature sets might prove useful.

It was desired, however, to assess the different features for their speaker discriminating ability independent of their speech modelling capabilities. For this reason, a standard framework for re-estimation of codebook weights was used across all feature sets. The framework used was the cepstra forward probabilities, which means that when re-estimating the codebook weights and the duration probabilities of any of the feature sets, the values of γ will come from the *cepstral* feature set.

For example, when Equation 3.11 is applied to obtain the Δcepstra codebook weights, values

of $P(\nu_m/x(l))$ from the *Δcepstra* feature set are used, but the values of $\gamma^k_{i,t,\tau}$ still come from the *cepstra* feature set.

3.4.9 Verification

Once speaker dependent models have been trained for a speaker those models can be used in a verification process to determine the *match* between an utterance and a model.

State Segmentation

The first stage of testing a client bid is referred to in this thesis as *state segmentation*. The goal is to allocate each frame to a state in a way that produces the most likely path through the state-time lattice.

We denote the state occupied during the t^{th} frame along the most likely path as q_t. The most likely path can then be completely described by a state sequence Q.

$$Q = q_1, q_2 \ldots q_T \tag{3.13}$$

The use of state-duration modelling does not allow the standard dynamic programming approach of the Viterbi algorithm to be used for state segmentation. This is because the Markov assumption of dependence only on the previous state has been broken.

If fixed transition probabilities are used in a strict left-to-right model the optimal path to state s_i at time t can be determined from the optimal path to state s_i at time $t-1$ and the optimal path to state s_{i-1} at time $t-1$. When state duration models are used, however, the optimal path to state s_i at time t also depends on the path from that state-time co-ordinate (i,t) to state N at time T in the state-time lattice (co-ordinate (N,T)).

A sub-optimal search procedure is used to find the state segmentation, in which the assumption is made that the optimum path to state s_i at time t depends only on the path up to time t, and not on the path after this time. The method is an adaptation of the standard Viterbi algorithm.

We define $\delta_i(t)$ as being the sub-optimal probability of being in state s_i at time t and in state j at time $t+1$. $\delta_i(t)$ is determined recursively as shown in Equation 3.17. We also define the state duration used to obtain $\delta_i(t)$ as $\xi_i(t)$ and it is derived simply as a by-product of the calculation of $\delta_i(t)$, as shown in Equation 3.18.

$$\delta_1(\tau) = d_1(\tau)\prod_{l=1}^{\tau} b_1(O_l), 1 \leq \tau \leq \tau_{max} \quad (3.14)$$

$$= 0, \tau > \tau_{max} \quad (3.15)$$

$$\xi_1(\tau) = \tau \quad (3.16)$$

$$\delta_i(t) = max_{\tau=1}^{\tau_{max}} \left[\delta_{i-1}(t-\tau) \times d_i(\tau) \prod_{l=t-\tau+1}^{t} b_i(O_l)\right] \quad (3.17)$$

$$\xi_i(t) = argmax_{\tau=1}^{\tau_{max}} \left[\delta_{i-1}(t-\tau) \times d_i(\tau) \prod_{l=t-\tau+1}^{t} b_i(O_l)\right] \quad (3.18)$$

For convenience we define the number of the first frame allocated to state s_i as Z_i which is determined recursively by a back-trace from Z_N as shown in Equation 3.19.

$$Z_N = T - \xi_N(T) \quad (3.19)$$

$$Z_i = Z_{i+1} - \xi_i(Z_{i+1}) \quad (3.20)$$

The state segmentation Q is trivially derived from the state durations using Equation 3.21.

$$q_t = i, \quad Z_i \leq t < Z_{i+1} \quad (3.21)$$

Verification Score

Note that any states which model silence are included in the state segmentation but are not included in the calculation of the verification score, because the quality of the match between the utterance and the silence model is not relevant to the speaker verification task.

Traditionally the log probability of the optimal state sequence, excluding silence states, is time-normalised by dividing by the number of non-silence frames and used as the verification score (Φ) to make an accept/reject decision. If s_1 and s_N are the silence states then traditionally the time normalised score Φ_{TNS} would be calculated as in Equation 3.22.

$$\Phi_{TNS} = \frac{\sum_{i=2}^{N-1}\left[\log(d_i(Z_{i+1} - Z_i)) + \sum_{t=Z_i}^{Z_{i+1}-1} \log(b_i(t))\right]}{\sum_{i=2}^{N-1}(Z_{i+1} - Z_i)} \quad (3.22)$$

In this thesis the state segmentation is used in a different way. Following the common

segmentation framework approach used in training, the LPC cepstra feature set is used to determine the optimal state-segmentation Q and this state segmentation is used to calculate verification scores for all feature sets.

It is the optimisation of this verification score which is the focus of the work in this thesis, and several alternatives are used.

Firstly the traditional Φ_{TNS} can be split into two components, one that is derived from the observation probabilities Φ_{OP} and another that is derived from the state duration probabilities Φ_{DUR}, as defined by Equations 3.23 and 3.24. Note that the duration score is divided by the number of non-silence states, rather than the number of frames.

$$\Phi_{OP} = \frac{\sum_{i=2}^{N-1} \sum_{t=Z_i}^{Z_{i+1}-1} \log(b_i(t))}{\sum_{i=2}^{N-1} (Z_{i+1} - Z_i)} \tag{3.23}$$

$$\Phi_{DUR} = \frac{\sum_{i=2}^{N-1} \log(d_i(Z_{i+1} - Z_i))}{N - 2} \tag{3.24}$$

Frame Weighting

A technique which is discussed in Chapter 5 is the use of frame weighting during the calculation of the verification score. The idea is that different frames have different speaker discriminating capabilities and that this can be exploited by weighting each frame according to its usefulness in the speaker verification task[6]. The time-normalisation is replaced by a division by the sum of the frame weights. The frame weights can be determined by various means. Whatever the basis for determining the weights, the principle is the same. The verification score using frame weights Φ_{FW} is given in Equation 3.25.

$$\Phi_{FW} = \frac{\sum_{i=2}^{N-1} \sum_{t=Z_i}^{Z_{i+1}-1} \omega_t \times \log(b_i(t))}{\sum_{i=2}^{N-1} \sum_{t=Z_i}^{Z_{i+1}-1} \omega_t} \tag{3.25}$$

The standard time-normalised score can be viewed as a simple form of frame weighting, using binary weights. When the time normalised score is calculated, frames which represent silence are not included in the calculation. The silence frames are, in effect, given a weight of $\omega = 0$, while all non-silence frames are given a weight of $\omega = 1$. The divisor is the number

[6] A frame weighting concept is used in the neural network system described in (Artieres & Gallinari, 1993).

of non-silence frames, which is the same as the sum of the frame weights. The framework for incorporating frame weights is therefore a general form of the standard time-normalised score.

3.4.10 Storage Requirements

This section examines the disk storage requirements of HASAS.

Codebook

In a likely application the codebooks would be stored centrally and shared between all the client speakers. The storage requirements are therefore not likely to be critical. They are small nonetheless.

Each of the 32 codewords in the 12 dimensional cepstral feature space has a 12 dimensional mean and a 12 dimensional diagonal covariance vector, giving a total of $32 \times (12 + 12) = 768$ floats per codebook.

Client models

Each client has models of each of the 12 digits (1-9, zero, nought, oh). Each model has 6 states each of which has 32 codeword weights and a duration mean and variance. This gives a total of $12 \times 6 \times (32 + 1 + 1) = 2448$ floats per feature set per client.

Assuming a 4 byte representation of floats, the storage requirement of HASAS using cepstra and delta cepstra feature sets is 19.6 kbytes per client, and 6.1 kbytes for the codebooks. For a thousand clients the total storage requirement of the system is 19.6 Mbytes.

This could, if necessary, be reduced by at least a factor of two by quantising the weights. Given the limited amount of data used to train the models, some granularity in the weights is unlikely to affect accuracy.

3.4.11 Decision Logic

If the scores from several digits from the same speaker are added together the resulting score provides a more robust basis for a verification decision than using a single digit score. We will refer to this as using *digit sequences*. A digit sequence is *not* a connected digit utterance. It is the concatenation of the results from several digits, each spoken in isolation. If the length of the digit

sequence is N then the sequence consists of the first N digits from the list {1, 2, 3, 4, 5, 6, 7, 8, 9, zero, nought, oh}. Various decision logics can be employed when using a digit-sequence.

1. Make a decision on each digit and then make an overall decision based on a majority vote.
2. Add the raw scores from all the digits and time-normalise the total score. This is equivalent to using a silence-separated string of digit models and applying them to the concatenated isolated digit utterances.
3. Add the time normalised scores for each of the digits and then apply a threshold to the total, or the average.

The first option should be less accurate than the other two because the process of making intermediate decisions throws away information. Was a digit strongly rejected or only narrowly? That distinction is lost in the majority vote.

The second option is probably the most intuitively appealing method because of its equivalence to modelling a concatenated sequence of digits directly, which is realistic to the application, so this method is used in this thesis. Earlier experiments using the third option have shown that there is no significant difference between the results obtained using the second and third options.

3.4.12 HASAS Overview

HASAS is intended for use in a telephone banking type application. The following constraints are placed on the system design.

- Isolated Digits.
- 5 Training Tokens per digit.
- Telephone speech from variable handsets.
- Real-time operation must be realistic.
- Storage requirements must be below 10kbytes per client.
- Concentration on initial performance of the system -no model adaptation is performed.

The models used are as follows.

- SCHMM.
- Gaussian state duration modelling.
- 6 state left-to-right word models.
- silence-digit-silence grammar.
- Multiple feature sets, each consisting of 32 diagonal covariance PDFs.
- No codebook re-estimation.
- LPC cepstra, Δcepstra, MFCC and ΔMFCC feature sets.

The key database features are as follows.

- Data collected in multiple sessions over 6 months.
- Speakers are native speakers of British English.
- 21 client speakers.
- 19 codebook speakers.
- 80 non-client impostor speakers.
- 2100 client speaker trials per digit.
- 10,500 impostor trials per digit.

3.5 Separating Speech and Speaker Modelling

The main distinguishing characteristic of the work in this thesis stems from a conceptual separation of speech and speaker modelling. The distinction is elucidated in this section, and the effects that this approach has on the architecture of HASAS are made clear.

There are significant similarities between the fields of speech and speaker recognition. Speech recognition research has a far greater community of researchers than speaker recognition, and for this reason speech recognition technology seems to have led speaker recognition. The most common approach to text dependent speaker recognition is to use a speech recogniser and rely on the inherent speaker dependence of the speech recogniser to perform speaker recognition.

While this is has proven to be a reasonable initial approach, it is fundamentally flawed, because while speech and speaker recognition are similar fields, they have some diametrically opposed goals. The objective and focus of a great deal of speech recognition research is to make the models and features speaker independent, *minimising* inter-speaker distance. The objective in speaker recognition is to find models and features which are strongly speaker dependent and which *maximise* inter-speaker distance while minimising intra-speaker distance.

A simple way to increase the chance of being falsely accepted by a *speech recognition* based ASV system is to speak clearly and precisely. This increases the likelihood score for the utterance because it matches the *speech* model well. If the speaker dependent model score is used directly for the verification decision, as is commonly done, the chance of false acceptance will be increased simply by improving the *quality* of the impostors speech, without necessarily increasing the impostor's similarity to the client speaker's voice. The traditional use of speaker dependent speech models for ASV is illustrated in Figure 3.5.

The probability of the Viterbi path through a speaker dependent HMM is a *combined speech and speaker recognition score*. The speech recognition component of this probability is a noise source in the speaker recognition task in the same way that the speaker recognition component is a noise source in the speaker independent speech recognition task.

The normalisation technique which was described in section 2.4.4 works by simply subtracting an estimate of the speech recognition component of the Viterbi path probability in order to gain a more robust estimate of the speaker recognition component. As was described previously, this technique is widely used with considerable success. Normalisation, however, does not address the fundamental problem (that speaker dependent speech models are combined speech and speaker models), it simply attempts to compensate for it.

The discriminating observation probability (DOP) technique described in Chapter 5 explicitly separates speech and speaker modelling. The DOP technique follows easily from the separation of the modelling process in the baseline system into a speech modelling stage followed by a speaker modelling stage.

The speech modelling stage consists of finding the Viterbi path through the HMM. This process is essentially that of state segmentation (optimally segmenting the speech frames into states). The way that this state segmentation information is used to produce a verification probability can then be considered the verification or *speaker modelling stage*.

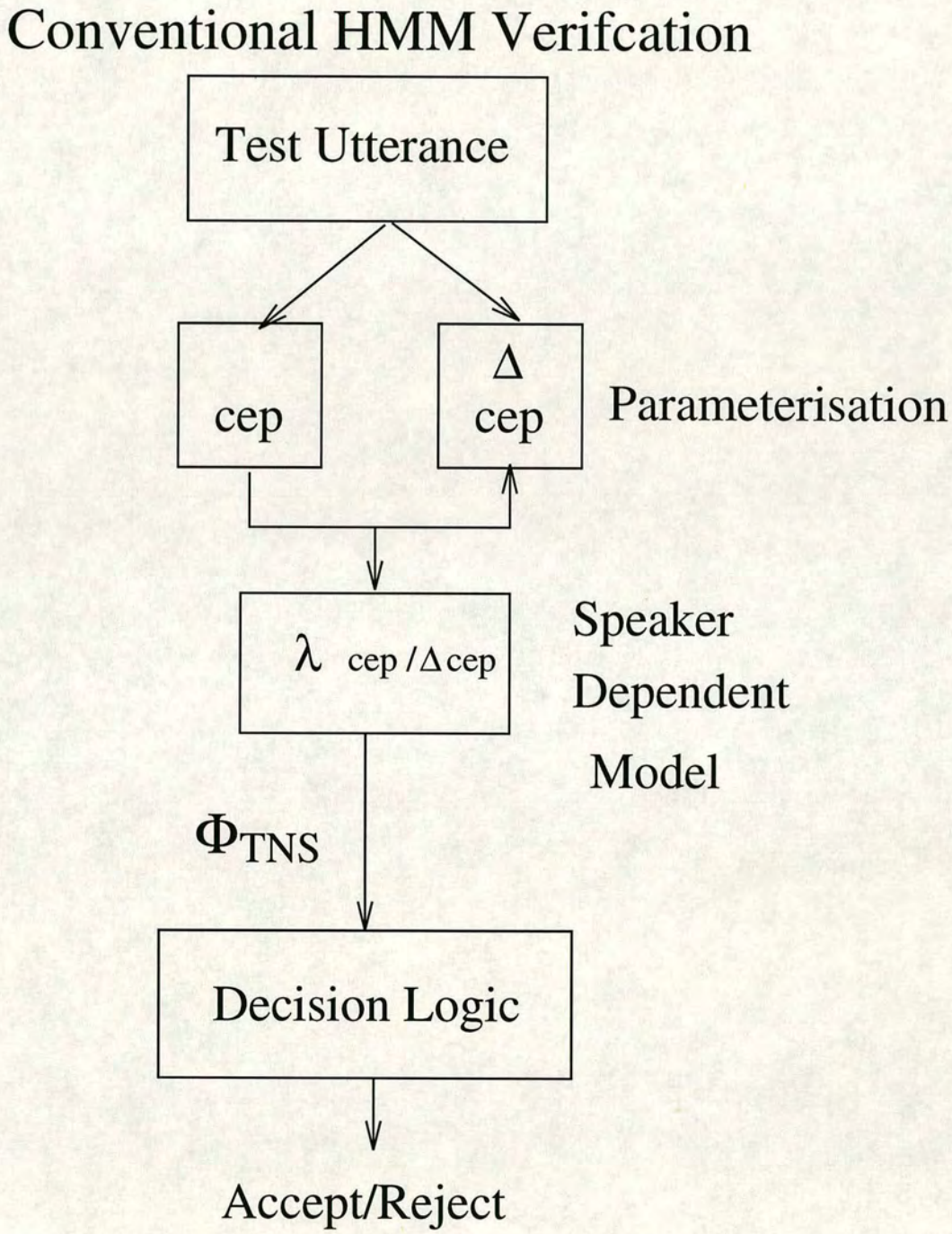

Figure 3.5: Block diagram of a traditional ASV system based on a speaker dependent speech recogniser.

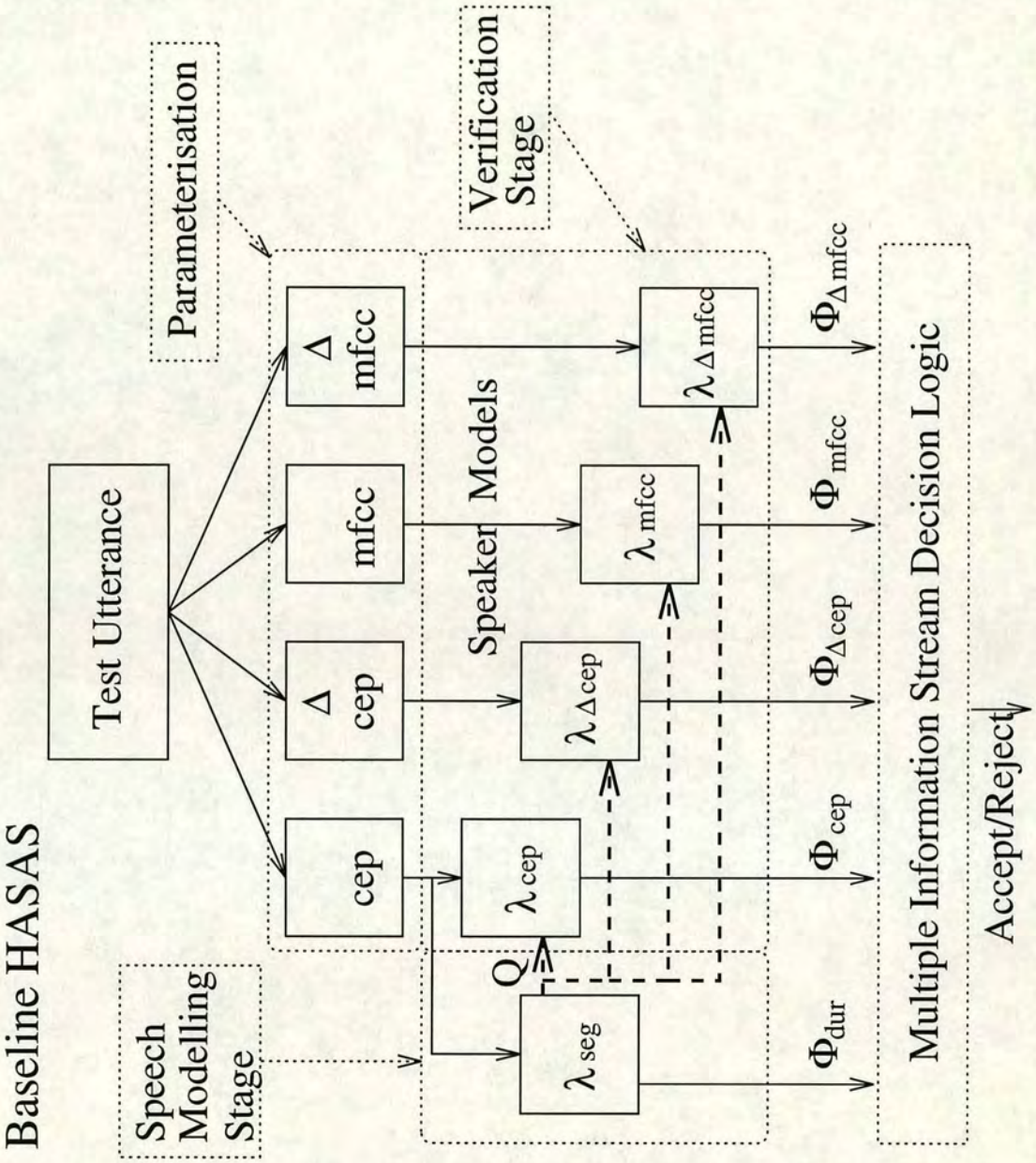

Figure 3.6: Block diagram of the pre-processing and client modelling modules of HASAS

Figure 3.6 shows a block diagram of the verifier used in the baseline HASAS. The unique feature of this system is the separation of speech recognition and speaker recognition blocks.

In this thesis the state segmentation is always performed using 12th order LPC cepstra based speaker dependent or speaker independent SCHMM with Gaussian state duration models. The work in the following chapters is concerned with finding the best features and algorithms for the speaker modelling stage -the verification calculation and decision.

The separation of the modelling process into speech and speaker modelling stages creates the opportunity for a divergence of speech and speaker recognition techniques. It is an original conceptual view of speaker recognition using HMMs and is the key to the motivation and success of the work described in this thesis.

Chapter 4

Evaluating HASAS

In the previous chapter HASAS was specified and the constraints used to arrive at this specification were identified.

In this chapter the performance of HASAS is evaluated. The theme of the work in this chapter is to make maximum use of the information available in the post-feature-extraction stage of the ASV process. The elements of the system which are investigated by experimentation are as follows.

- The relative performance of the digits (Section 4.1).
- The effect on performance of using more than one digit (Section 4.1.1).
- The difference between using speaker specific and speaker independent thresholds (Section 4.2).
- Weighting the scores from each of the digits according to how useful the digit is in the ASV task (Section 4.3).
- The relative performance of several commonly used speech feature sets in the *speaker modelling* stage (Section 4.5).
- The effect on performance of using more than one feature set in a multiple codebook system (Section 4.7).
- The use of state duration probabilities as an additional information source in the verification decision (Section 4.8).

A detailed analysis of the distribution of verification errors across client and impostor populations is performed in Section 4.10.

Digit	Threshold τ_{EER}	SI EER
1	-1.99	16.91
2	-2.08	14.90
3	-1.93	16.58
4	-1.55	24.60
5	-2.10	13.12
6	-2.12	15.89
7	-2.16	11.82
8	-1.98	15.36
9	-2.12	11.73
zero	-2.12	13.08
nought	-1.98	17.26
oh	-2.05	14.32

Table 4.1: EER and threshold for each of the 12 digits using a SI threshold (LPC Cepstra)

4.1 Single Digit Performance: LPC Cepstra

The first stage in evaluating HASAS is to evaluate the performance on a single isolated digit test utterance. The cepstral feature set was used for this evaluation. The verification score used is Φ_{OP} and the decision logic consists of applying an equal error rate threshold τ_{EER} on Φ_{OP}. τ_{EER} was speaker independent but digit specific. The use of EER thresholds means that all thresholds are determined *a posteriori*.

$$\Phi_{OP} \geq \tau_{EER} \Rightarrow \texttt{accept} \tag{4.1}$$

$$\Phi_{OP} < \tau_{EER} \Rightarrow \texttt{reject} \tag{4.2}$$

The EER was evaluated for each of the 12 digits. The performance of each of the digits and the thresholds used are listed in Table 4.1 and the EERs are compared graphically in Figure 4.1.

There is considerable variation in performance among the digits. The average is 15.5% with a range from 11.7% to 24.6%.

The differences in EER could well be due to differences in the amount of speaker discriminating information in the various digits. If this is true it could be used to advantage. This possibility is investigated in Section 4.3.

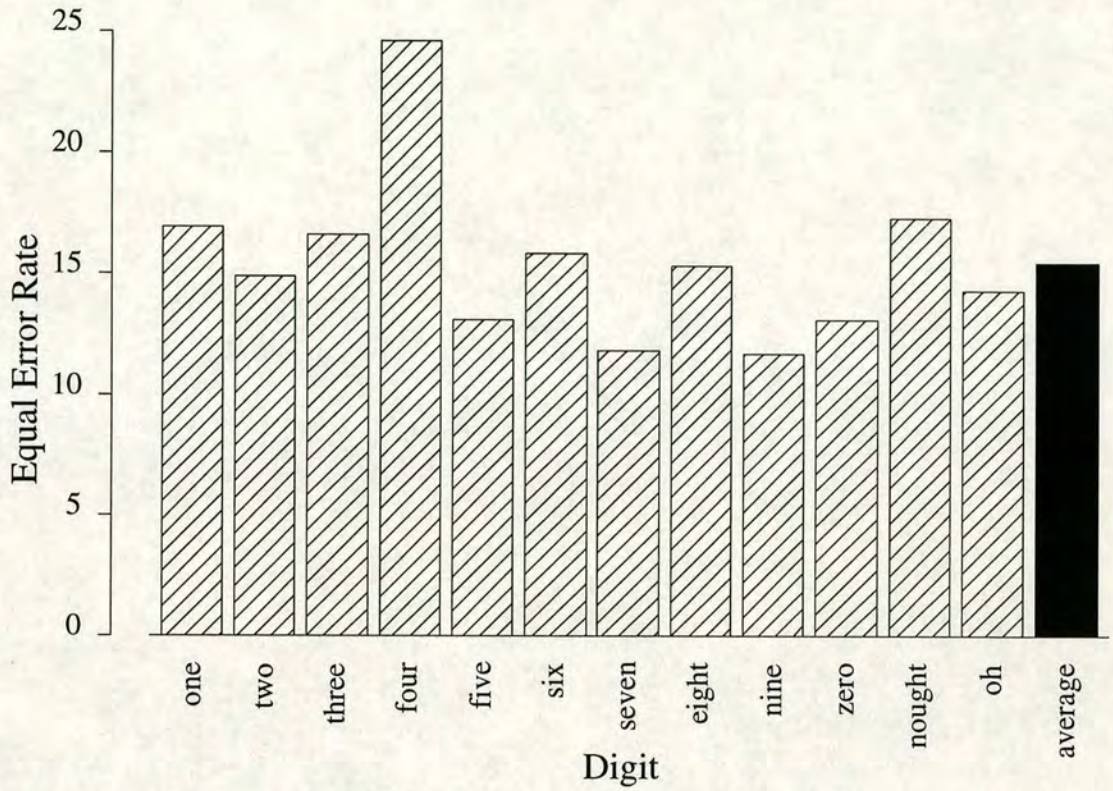

Figure 4.1: EER for each of the 12 digits (LPC Cepstra)

4.1.1 Digit Sequence Performance

While the probability scores for the different digits are not completely independent in the probabilistic sense, it would be expected that the information in the various digits (and, to a lesser extent, different utterances of the same digit) is partially uncorrelated. Combining the scores from several digits could therefore be used to improve performance. This has been supported by studies in the literature, for example (Rosenberg *et al.*, 1991).

The verification score Φ_{OP} is Equation 3.23 extended over multiple digits and is defined in Equation 4.3, where W is the number of digits in the sequence and the superscript l denotes the l^{th} digit.

$$\Phi_{OP} = \frac{\sum_{l=1}^{W} \sum_{i=2}^{N-1} \sum_{t=Z_i^l}^{Z_{i+1}^l - 1} \log(b_i^l(t))}{\sum_{l=1}^{W} \sum_{i=2}^{N-1} (Z_{i+1}^l - Z_i^l)} \tag{4.3}$$

If the scores from several digits from the same speaker are added together the resulting score provides a more robust basis for a verification decision than using a single digit score. This is referred to here as using *digit sequences*.

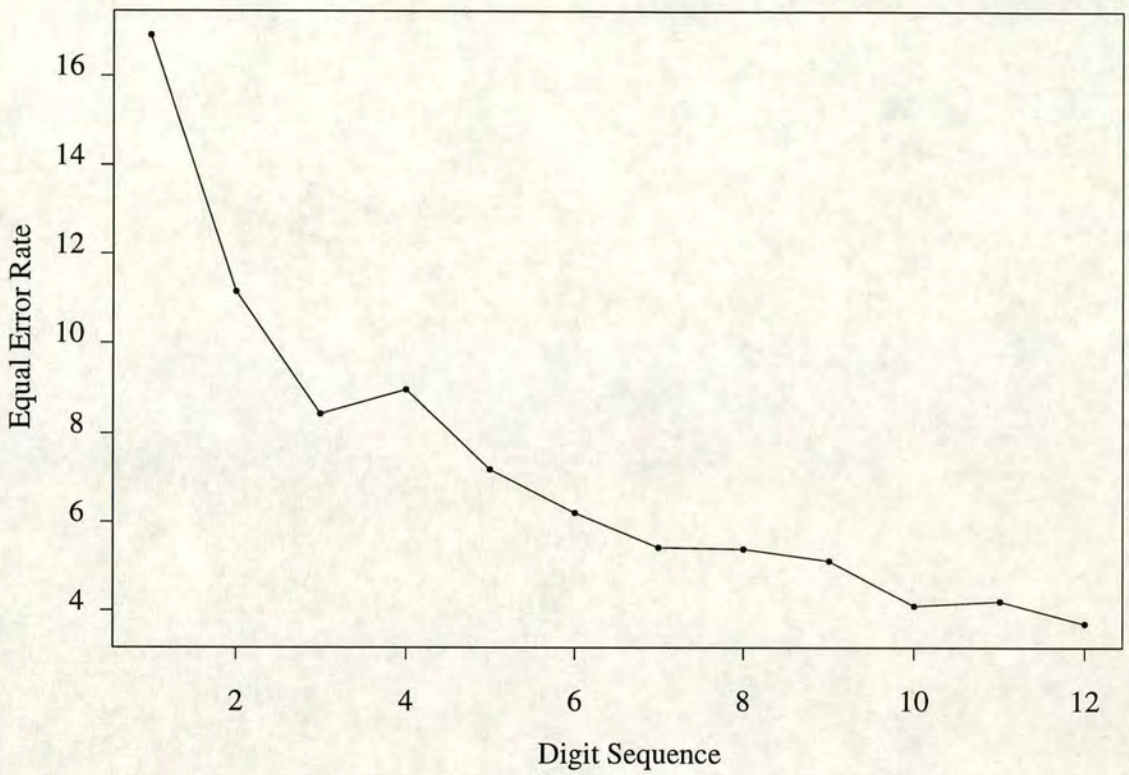

Figure 4.2: EER for various digit sequence lengths.(LPC Cepstra)

A digit sequence is *not* a connected digit utterance. It is the concatenation of the results from several digits, each spoken in isolation. If the length of the digit sequence is N then the sequence consists of the first N digits from the list $\{1, 2, 3, 4, 5, 6, 7, 8, 9,$ zero, nought, oh$\}$.

A single threshold is applied to the verification score taken over all the digits in order to make the verification decision. This has the same effect as if a concatenation of silence-separated word models was used on a concatenation of isolated digit utterances.

The EER for various sequence lengths are compared in Figure 4.2. Note that a digit sequence of length 1 consists only of the digit *one*, and a two-digit-sequence consists of the digits *one* and *two*, and so on.

It can be seen from the plot of EER that the addition of digits increases the performance almost monotonically[1]. The increase in EER of the four-digit-sequence compared to the three-digit-sequence is due to the addition of the digit *four*. Recall from from Table 4.1 that this digit had by far the worst performance. The correlations in the speaker discriminating information of the digits eventually become apparent as more digits are added and the benefit of adding digits

[1] If all combinations of digits were tested exhaustively, we would expect a monotonic improvement in performance

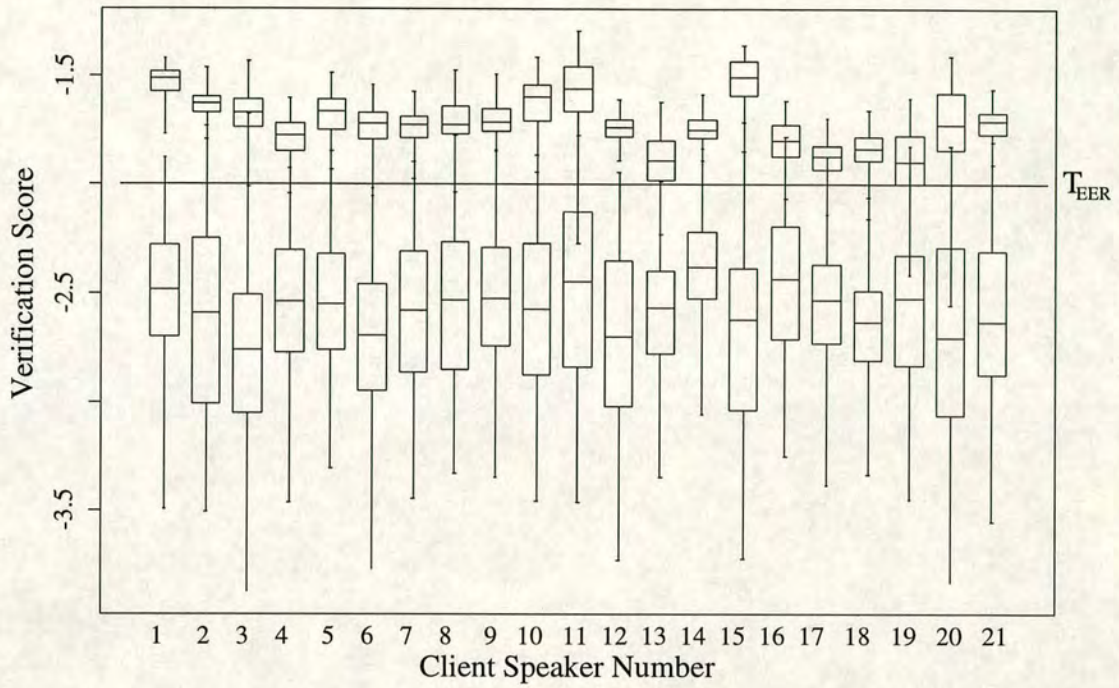

Figure 4.3: Box-plot of the genuine speaker score distribution and the impostor score distribution for each of the speakers. The top box is the genuine speaker distribution and the bottom box is the impostor score distribution. The box represents the second and third quartiles and the line in the box indicates the mean. The whiskers show the extremes of the distribution. The speaker independent EER threshold is shown by a solid line. The scores are from the 12-digit string (LPC Cepstra)

appears to tail off at around 10 digits.

4.2 Speaker Specific Thresholds

The setting of thresholds is not a trivial task. The use of an EER means that the threshold is determined *a posteriori* which, of course cannot be done in practice. In the same way that it is preferable to use different thresholds for each digit, it is possible that it would be advantageous to use a different threshold for each speaker.

Figure 4.3 is a box-plot of the distribution of genuine and impostor score distributions for each of the speakers (from the 12-digit sequence). The speaker independent EER threshold is shown as a solid line. It can be clearly seen that this threshold is sub-optimal for several speakers. Speakers 1 and 12, for example, would both have no errors if a speaker specific threshold was used. Many other speakers such as speakers 2, 5, 7, 9, 14, and 15 would produce far fewer errors

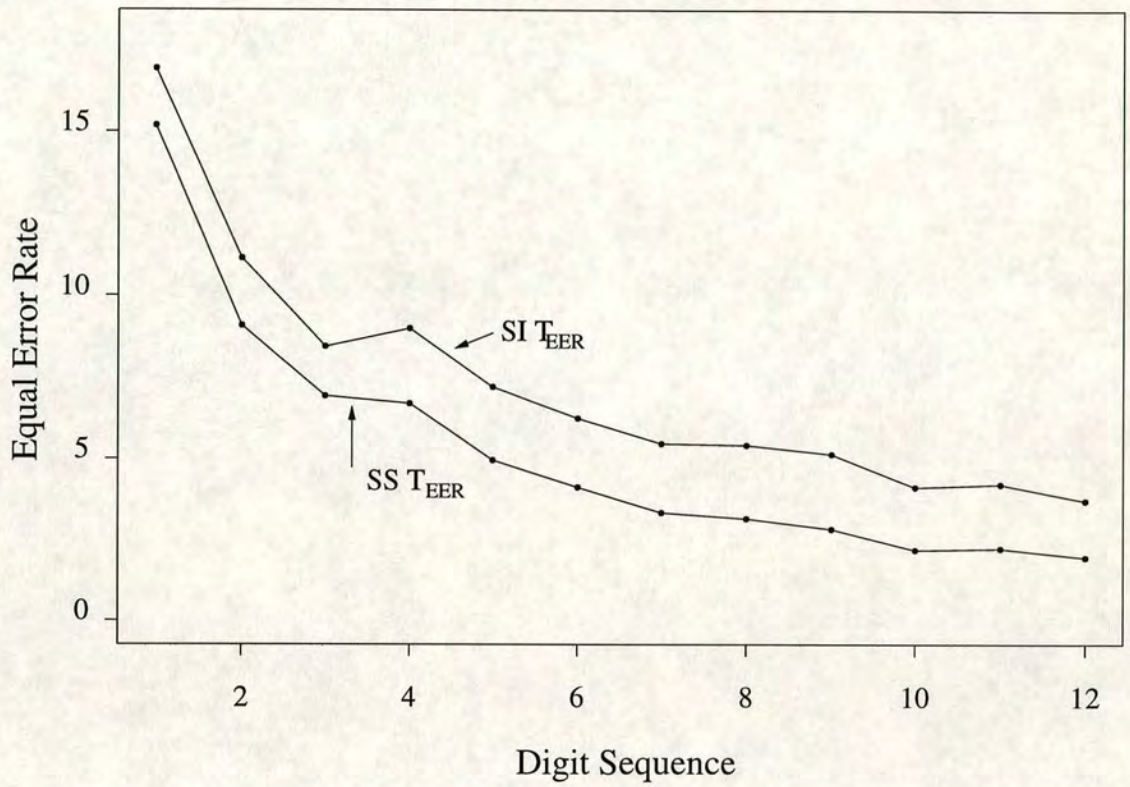

Figure 4.4: The improvement of speaker specific thresholds over speaker independent thresholds for various digit sequences. The average decrease in error is 34% (LPC Cepstra).

if a different threshold were used.

Figure 4.4 shows the improvement gained from the use of speaker specific thresholds. The EER reduction ranges from 16% for a single digit to 48% for the 12-digit string. It is therefore very important to make a clear distinction between *speaker specific* and *speaker independent* thresholds when reporting results.

Several studies in the literature use speaker specific thresholds without any explanation of how these thresholds would be determined. Many studies do not even give sufficient information to judge whether the EER thresholds are speaker specific or not. The difficulty with determining speaker specific thresholds *a priori* is that testing on the training data does not give an accurate estimate of likely client scores. Scores from training data (closed-test scores) are artificially high, because the data are well modelled. Extra data could be requested at enrolment that could be set aside for threshold estimation but since the amount of enrolment data that can be reasonably requested is strictly limited, this is unlikely to be possible. If a way can be found to estimate speaker specific thresholds without increasing the amount of enrolment data required, the benefit

is clear.

One method of estimating speaker specific thresholds is to jack-knife the training data to obtain some open test client verification scores, and use some impostor speakers to obtain some impostor verification scores. These scores can be used to adapt the speaker independent thresholds to make them more speaker specific. Further data can be gathered as the system is used to further adapt the thresholds.

The results for the LPC cepstra based system using both speaker independent (SI) and speaker specific (SS) thresholds are summarised in Tables B.1 to B.3.

4.3 Weighted Digit String

It was noted previously in Section 4.1 that the speaker discriminating performance varies considerably from digit to digit. This is almost certainly determined in some way by the phonetic makeup of each digit. Several studies have been done which attempt to rank the usefulness of different phonemes for the speaker recognition task (Eatock & Mason, 1994; Eatock, 1992; Mokhtari & Clermont, 1994; Nolan, 1983; Floch *et al.*, 1994; Parris & Carey, 1994). Other systems use a voiced/unvoiced classification to select speech that is useful for ASV (Matsui & Furui, 1991; Lipeika & Lipeikiene, 1993). In (Savic & Gupta, 1990), an initial segmentation using an HMM is followed by a series of models of different broad phonetic classes. The scores from the models were weighted according to their speaker discriminating power.

A reasonable hypothesis is that if each of the single digit verification scores is weighted according to the speaker verification performance of that digit then the digit sequence scores should improve. This hypothesis is tested in this section.

The weights for each of the digits are calculated according to Equation 4.4. They are speaker independent since the same weights are used for each speaker.

$$\varpi_i = 1 - c \times \frac{EER_i - \min_i(EER)}{\max_i(EER) - \min_i(EER)} \tag{4.4}$$

Table 4.2 shows the weights for each of the digits, based on the single digit EER using speaker specific thresholds. The normalisation with $c = 0.5$ sets the digit with the best EER to have the weight of 1.0 and the digit with the worst EER to have the weight of 0.5.

Digit	Weight	SS EER
7	1.00	9.9
9	0.99	10.1
zero	0.96	10.9
oh	0.91	11.9
5	0.90	12.1
8	0.87	12.9
2	0.87	13.0
6	0.82	14.1
3	0.82	14.2
nought	0.78	15.1
1	0.77	15.2
4	0.50	21.6

Table 4.2: Normalised weightings of each of the digits based on single digit EER performance using speaker specific thresholds.

It is hypothesised that weighting the verification scores of each of the digits by its ranking before adding them to obtain the 12-digit string score should improve the overall performance. The verification score has the form of Equation 4.5.

$$\Phi_{OP} = \frac{\sum_{l=1}^{W} \varpi_i \times \sum_{i=2}^{N-1} \sum_{t=Z_i}^{Z_{i+1}-1} \log(b_i(t))}{\sum_{l=1}^{W} \varpi_i \times \sum_{i=2}^{N-1} (Z_{i+1} - Z_i)} \tag{4.5}$$

The EER obtained for the digit sequences using weighted digits are compared with those obtained without using weights in Figure 4.5.

It can be seen that the speaker independent weights provide no useful improvement in the EER. Two explanations are offered for this. Firstly it possible that while the digits have different performance on their own, each digit contains different speaker discriminating information and all the digits are equally important, and weighting therefore serves no purpose. Another possibility is that the rankings of the digits varies from speaker to speaker and using speaker specific weights may improve the technique.

4.4 Speaker Specific Digit Weights

If the weighting for each digit is calculated separately for each speaker it can be seen that there is considerable variation from speaker to speaker. Figure 4.6 is a box-plot which shows the range

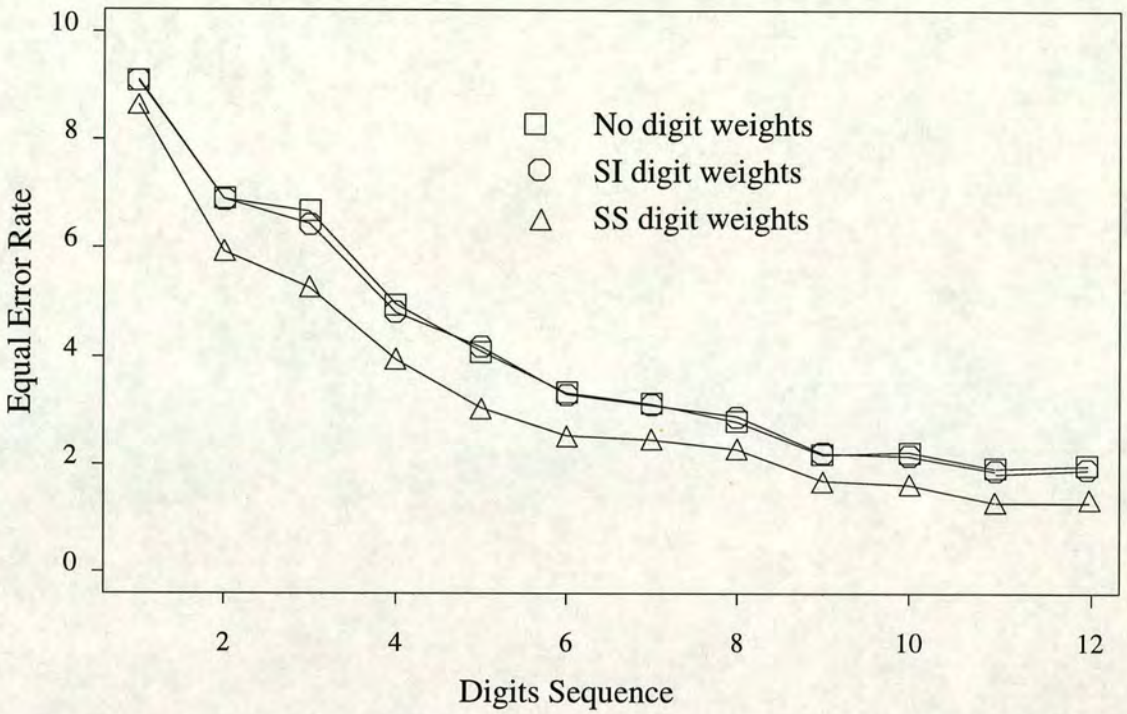

Figure 4.5: Performance of the three algorithms for various digit sequences. (a) No digit weighting (b) Speaker independent digit weights c = 0.5. (c) Speaker specific digit weights c = 0.5. (All EER are calculated using speaker specific thresholds on LPC cepstra models).

of weights obtained by each digit over all the speakers. The top of the graph represents a weight of 1.0 which means that that the digit had the best EER of all the digits, the bottom of the graph corresponds to a weight of 0.5 which means that the digit had the worst EER of all the digits.

There is considerable variation in the relative performance of digits from speaker to speaker. Note, for example, that although the digit *zero* performed well for most speakers it was the worst for one speaker, likewise for the digit seven.

Figure 4.5 shows the effect of using speaker specific digit weights. Whereas the speaker independent weights produced no useful advantage the speaker specific weights do. Several values of the constant c were tried. The best was c = 1 which produced a 33% drop in EER on the 12 digit sequence (from 1.93% to 1.29%). Increasing the value of c increases the influence of the weights until at c = 1 the worst digit is eliminated from the sequence. This result means that in general at least one of the digits is not worth using -but that that digit varies from speaker to speaker.

The data from the digit weight experiments are summarised in Table 4.3.

In Figure 4.7 the effect of the digit weighting can be examined in more detail. This graph

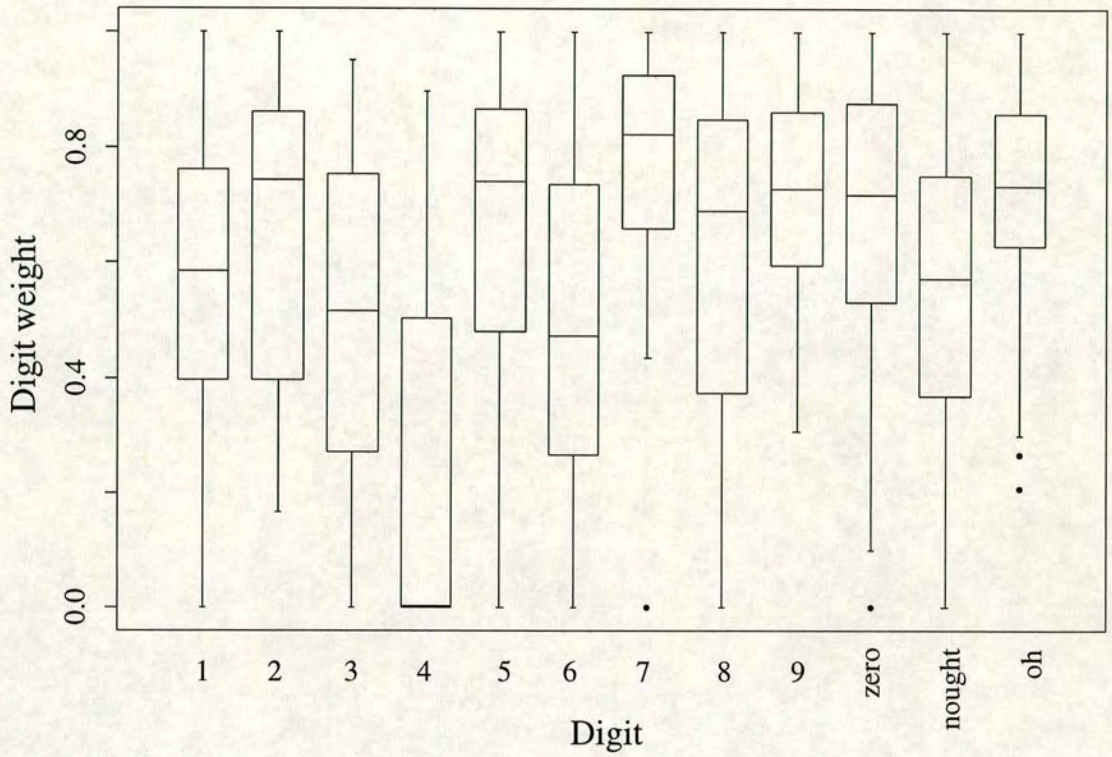

Figure 4.6: Box-plot of speaker specific weights. Each box-plot consists of the weights for a particular digit over all the speakers. (LPC Cepstra). Note that the horizontal line in the middle of each box represents the mean over all speakers for that word, and the upper and lower edges of the box represent the upper and lower quartiles. Dots represent outliers corresponding to individual speakers.

	SS EER
No weights	1.93
SI weights(c=0.5)	1.88
SS weights (c=0.5)	1.59
SS weights (c=0.66)	1.44
SS weights (c=1)	1.29

Table 4.3: EER performance of 12-digit sequence with: (a) No digit weighting (b) Speaker independent digit weights. (c) Speaker specific digit weights. (All EER are calculated using speaker specific thresholds)

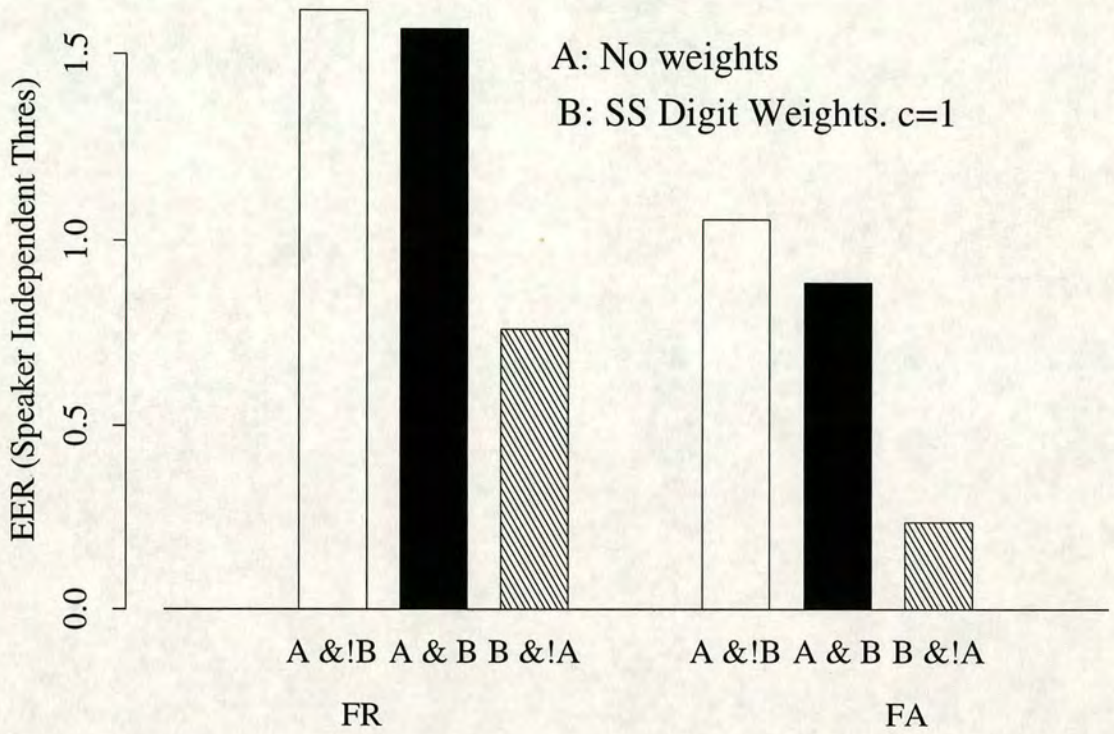

Figure 4.7: Bar-graph showing a comparison of errors created and eliminated by using speaker specific digit weights with c = 1. The eliminated errors are in white , the created errors in grey and the unchanged errors in black. The two sets of bars represent the FR and FA errors. (LPC Cepstra)

shows the errors created and eliminated by the digit weighting relative to using no weights. It can be seen that while the weighting does create some errors (grey bars), it eliminates far more (white bars).

While there is clearly an advantage in using speaker specific digit weights, the weights must first be determined for each speaker. This brings up the same difficulties encountered in the estimation of speaker specific thresholds and the same solution is suggested. An initial estimate can be made by jack-knifing the training data and the estimate is then refined while the system is in use.

The fact that speaker specific weights can substantially improve performance indicates that utterances could potentially be found which were specifically tailored to suit each client speaker.

4.5 Comparing Feature Sets

The feature set chosen as the basis for a speaker verification system is very important. Ideally such a feature set should capture and emphasise all aspects of the speech signal which show large inter-speaker distance and discard anything which shows little inter-speaker distance or large intra-speaker variation.

This means that while ASR and ASV have much in common, some aspects of the two techniques are fundamentally opposed. It is somewhat surprising, then, that the feature sets generally used for ASV are much the same as those used for ASR, since the same feature set is very unlikely to be ideal for both. The choice of feature sets is a case of selecting from a list of imperfect options.

The results presented so far come from using the LPC cepstra feature set. In this section we evaluate the performance of three other widely used feature sets. These are mel-frequency cepstral coefficients (MFCC) and the difference coefficients of both cepstra and MFCC, denoted Δcepstra and ΔMFCC respectively.

What is of interest here is which features have the most speaker discriminating information. In order to isolate this as much as possible, the speech recognition stage of the verification process, namely the state segmentation, was made common to all experiments. This is illustrated in Figure 3.6. The LPC cepstra feature set was used for the state segmentation. There is no special reason for this choice, nor is there any need for only one feature set to be used.

This separation of speech and speaker modelling is a very important point. Feature sets which are optimised for speech recognition can be used to perform state segmentation and feature sets which are optimised for speaker discrimination can be used to calculate the verification score. Leaving aside which feature set is best for state segmentation (which is an ASR problem), this section looks at which feature sets are the best for speaker discrimination.

4.5.1 Results

The EER performance for each of the features over various digit sequence lengths is shown in Figure 4.8. Tables B.1 to B.12 contain the detailed summaries for each feature set. The relative performance of the feature sets for the 12 digit sequence is given in Table 4.4.

It is possible that the fact that the segmentation is performed using LPC cepstra improves the

Feature	SI Threshold	SS Threshold
	EER	EER
cepstra	3.69	1.93
Δcepstra	9.75	4.4
MFCC	5.08	2.59
ΔMFCC	17.84	9.53

Table 4.4: 12 digit sequence results for 4 different features. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers.

performance of that feature set over others, but it is unlikely that this has a great effect since the state segmentation provided is likely to be reasonable for all features. Further experimentation would be needed to clarify this.

Figure 4.9 compares the relative rankings of the digits for the four feature sets. In general the relative performance of the digits follows the same pattern over the four feature sets. Exceptions to this are that the digit one ranks higher with the cepstra feature sets than with the MFCC feature sets, and that the digits *one* and *oh* rank lower for the ΔMFCC feature set than the Δcepstra feature set. There are a number of factors influencing these results. Firstly, as has already been established, the relative importance of the digits varies from speaker to speaker. Secondly, as will be seen in Section 5.4, the performance of the different *features* varies from speaker to speaker. Finally, the variation in the rankings of the digits is due to both the *quality* and the *quantity* of speaker discriminating information they contain.

Different speakers will show different levels of intra and inter-speaker variability in the different areas of feature space, and the different feature sets will enhance different types of variability for a given area of feature space. Assuming that each state represents a single acoustic event, occupying a particular area of feature space, weights should be not only speaker and feature-set specific but also *state* specific, in order to maximise the use of the available speaker discriminating information. This would be an interesting area for further research.

4.6 Multiple Feature Sets

The previous section established the relative performance of the feature sets. Rather than choosing the best feature set from these, this section explores the option of using two or more

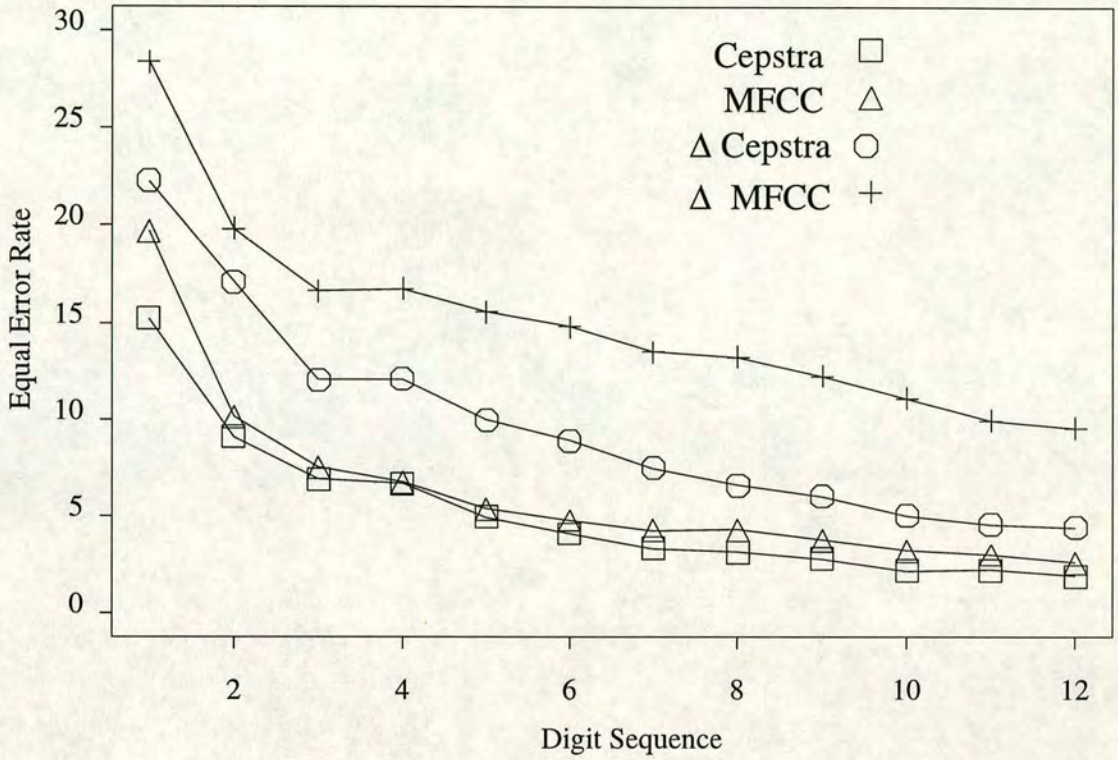

Figure 4.8: Performance over various string lengths of the four feature sets. a) LPC cepstra b)MFCC c) Δcepstra d) ΔMFCC. These results are for a speaker specific (SS) threshold.

feature sets in order to improve the robustness of HASAS.

4.6.1 Combining Multiple Feature Sets

The most obvious way to use multiple feature sets is to have a collection of models, each of which uses a different feature set. The disadvantage of this approach is that the computation involved can increase considerably.

The feature extraction stage can become much more computationally expensive depending on how much computation the feature sets have in common. For example, difference cepstra can be easily derived from LPC cepstra but fast-Fourier transform (FFT) based features such as MFCC cannot, and if LPC cepstra and MFCC were used together the computational cost of the feature extraction stage would roughly double.

The computation in the post-feature-extraction modelling process is proportional to the number of models, because a Viterbi search must be performed for each model. In this work this potential computation increase is eliminated by employing a single Viterbi search and using

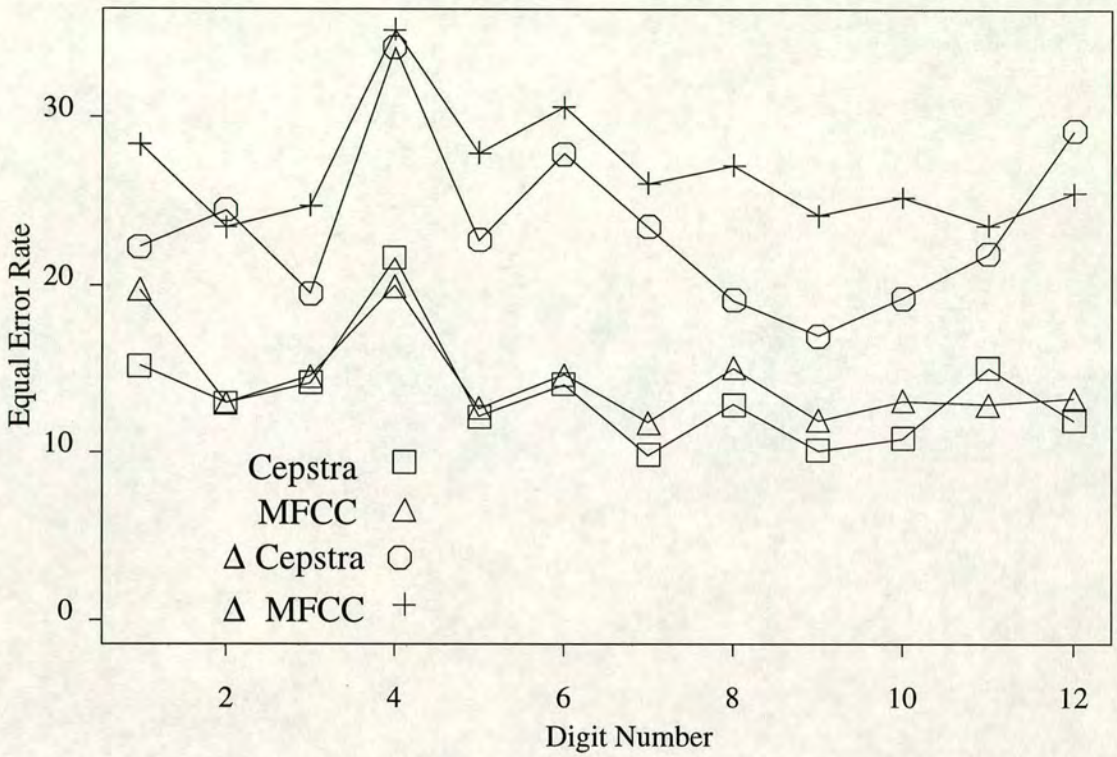

Figure 4.9: Relative performance of single digits for four different feature sets. From this graph it can be seen whether the relative rankings of the digits are the same for all features. The EERs are calculated using a speaker specific threshold.

the state segmentation it produces in each of the feature models (refer to Figure 3.6).

A similar approach was taken in the training during Baum-Welch re-estimation (Section 3.4.8). Note that the reduction in computation is a beneficial side-effect of the approach taken here, rather than the goal, which was the separation of speech and speaker modelling. As in the single feature set experiments, the LPC cepstra feature set is used to perform the state segmentation.

Multiple feature sets give multiple *information streams* from which to make the verification decision. This is not necessarily better, nor is the optimum combination of features immediately apparent, because the usefulness of a combination of features depends on the independence of the speaker discriminating information they contain.

4.7 Pair-wise combinations of information streams

If the information contained in the features used to produce two different verification scores is sufficiently uncorrelated, it is likely that a combination of the scores or *information streams* could produce a more robust probability than either score alone.

Figure 4.10 is a plot of the verification score using cepstra against the verification score using Δcepstra.

The scores from each feature have been divided by the variance of that feature model's client scores so that the relative importance of the two features can be directly compared. The use of the cepstra feature set alone corresponds to using a vertical line as the decision threshold. Likewise the use of the Δcepstra score alone corresponds to the use of a horizontal line threshold.

It can be seen from this plot that the client and impostor speaker clusters can be best separated using a diagonal line as a threshold , rather than either a vertical or horizontal line. This means that the two measures have partial independence of information and that a combination of the two scores will improve speaker verification performance.

4.7.1 Combining Verification Scores

The combination of the two verification scores follows an approach similar to that used by (Soong & Rosenberg, 1988) to combine cepstra and Δcepstra distances from two VQ codebooks.

The normalised verification scores are combined in a simple weighted sum as shown in Equation 4.6.

$$\Phi(\mathrm{pair}) = \alpha \times \Phi(\mathrm{cepstra}) + (1 - \alpha) \times \Phi(\Delta\mathrm{cepstra}) \tag{4.6}$$

For each pair of information streams, values of α between 0.0 and 1.0 in steps of 0.1 were used to obtain the best combination. Varying α corresponds to using diagonal lines of varying gradients in Figure 4.10. This simple linear combination is of course a first approximation to an optimal combination. The use of non-linear combinations of information streams, which produce curves as thresholds in the two-dimensional feature space, is likely to be a fertile area for future research.

Figure 4.10: Scatter-plot of cepstra verification score against Δcepstra verification score. The client and impostor clusters are clear, and it can be seen that the combination of the two scores provides a better decision space for classification that either score alone. The impostor scores are represented by commas and the client scores by dots.

4.7.2 Results for Pair-wise Combinations of Feature Sets

Figure 4.11 shows the 12 digit sequence results when pairs of feature sets are combined, using a speaker specific threshold.

In many cases the performance is reasonably robust to the choice of α. In general the cepstral features should be weighted more than the MFCC features and the static features should be weighted more than the difference features.

Table 4.5 gives the EER for several pair-wise combinations of verification scores. The best value of α is used and is quoted in the results tables. It can be seen that in all cases there is a reduction in error rate from combining the verification scores from two feature sets. These results are discussed in the remainder of this section.

1^{st} Feature Set	2^{nd} Feature Set	SI Threshold			SS Threshold		
(α)	($1-\alpha$)	α	EER	Reduct.	α	EER	Reduct.
cepstra	Δcepstra	0.7	3.08	16%	0.6	1.36	30%
cepstra	MFCC	0.9	2.92	21%	0.7	1.39	28%
cepstra	ΔMFCC	0.8	3.44	7%	0.7	1.67	14%
Δcepstra	MFCC	0.7	3.35	34%	0.8	1.65	36%
Δcepstra	ΔMFCC	0.8	9.38	4%	0.7	4.25	4%
MFCC	ΔMFCC	0.6	4.37	14%	0.5	2.22	14%

Table 4.5: Pair-wise feature set results. EER for 12 digit string. The value of α gives the relative weightings of the two information streams. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers. *Reduct* is the percentage reduction in error rate gained by using the pair instead of using the better feature set on its own.

4.7.3 Addition of Delta Feature Set Model

The difference or Δcepstral features have been successfully used in speaker recognition systems (Soong & Rosenberg, 1988; Furui, 1981) to provide complementary information to the static coefficients.

Difference coefficients encode the changes in the spectral characteristics of the speech and as such provide complementary information to the *static* features. Often noise sources appear as biases in the static features such as cepstra, giving rise to such techniques as cepstral mean subtraction (Rosenberg *et al.*, 1994). Difference features are by their nature immune to such slowly varying biases, which increases their attractiveness as a feature set in noisy environments.

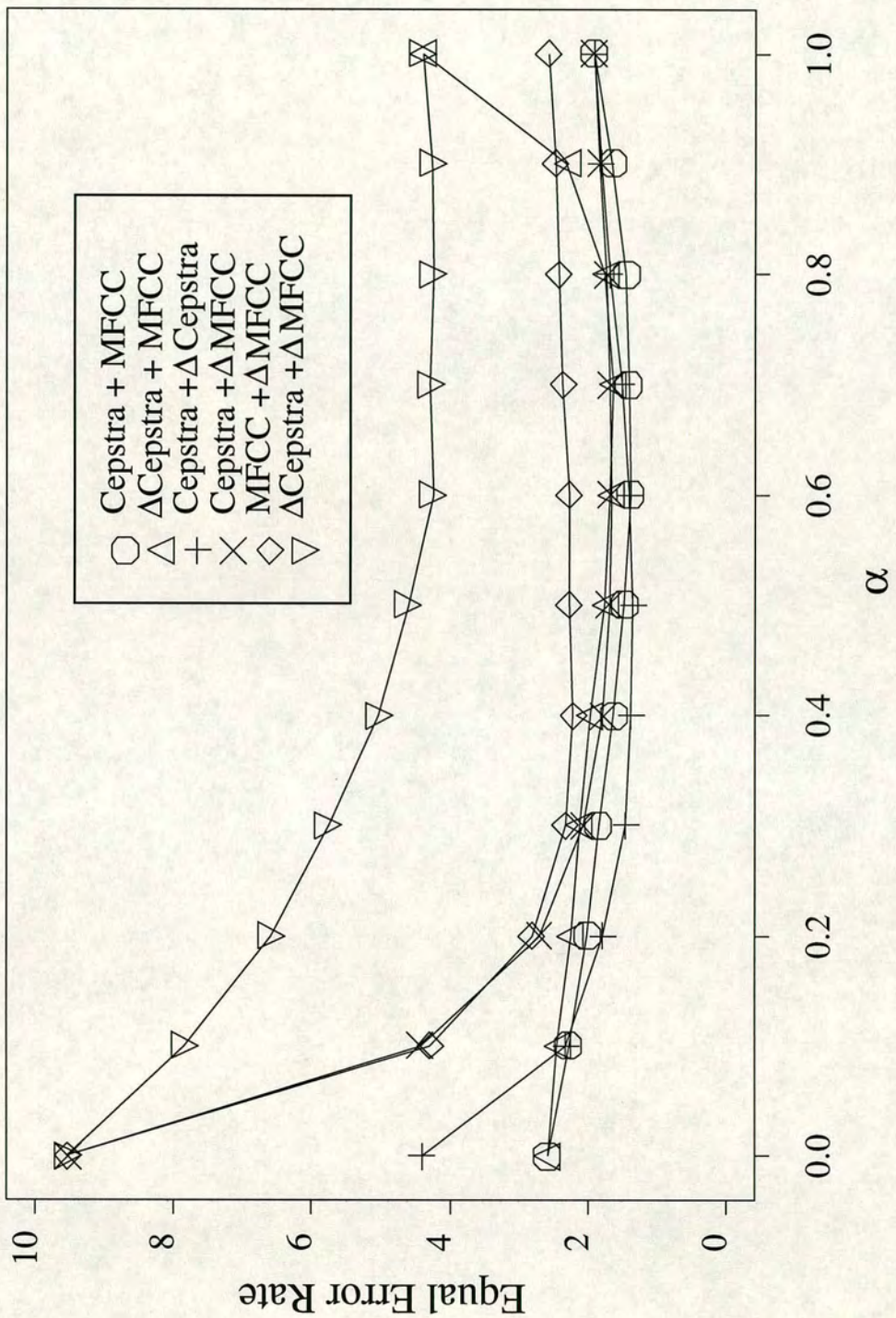

Figure 4.11: Performance over various string lengths of the six pair-wise combinations of verification scores from the different feature models. These results are for a speaker specific (SS) threshold.

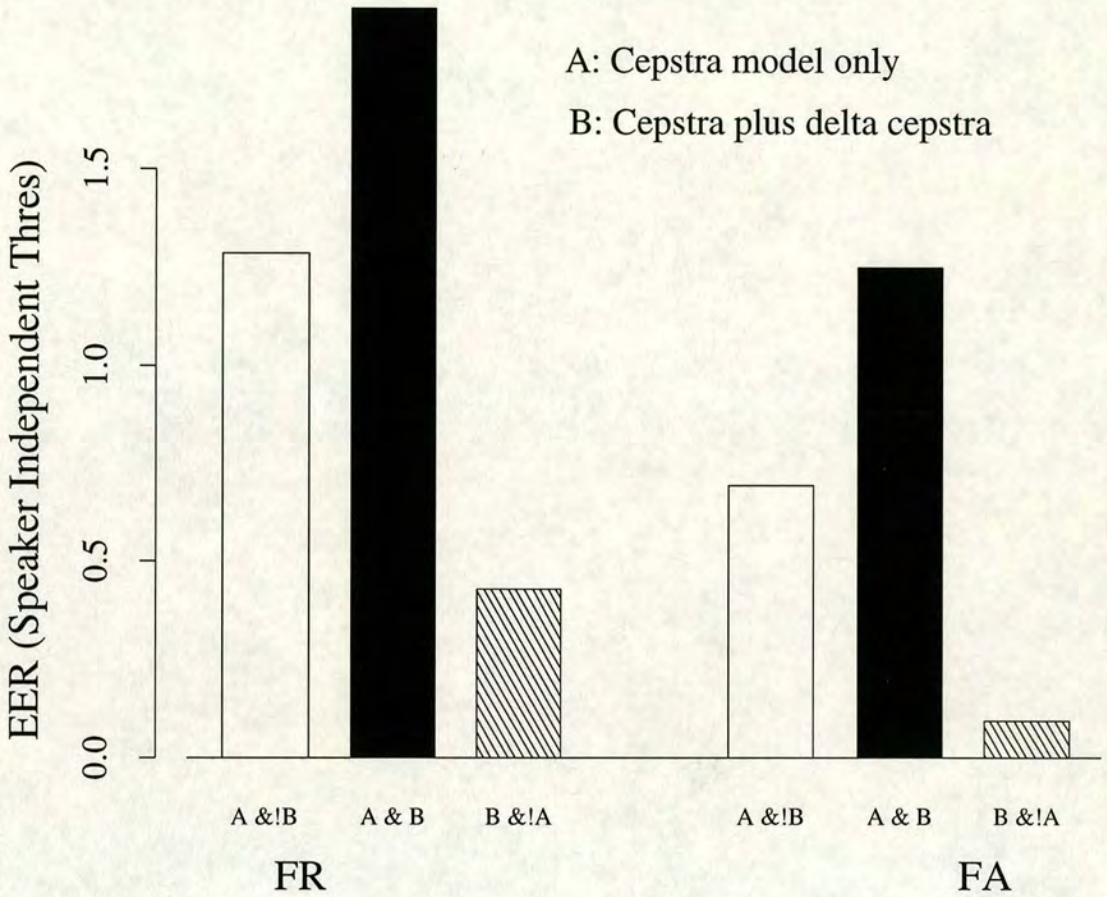

Figure 4.12: Bar-graph showing a comparison of errors created and eliminated by using the delta cepstra model scores in a weighted linear combination with the cepstral scores, relative to just using the cepstral scores. The eliminated errors are in white, the created errors in grey and the unchanged errors in black. The two sets of bars represent the FR and FA errors. (SI EER Thresholds).

The results in Table 4.5 support the hypothesis that static and dynamic features contain partially independent information. Substantial EER reductions of 7-36% are gained from the static-delta combination of both cepstra and MFCC. The best results are obtained by combining cepstra with Δcepstra.

A summary of the results for the four static plus delta combinations, using the optimal value of α in each case is given in Tables B.16 to B.33. The errors that are eliminated by adding the Δcepstra model to the cepstra model are analysed in figure 4.12. It can be seen that while the use of Δcepstra does cause some new errors, it eliminates far more.

Recall that the static plus Δ combinations of the same feature add little to the computational load. The Δcepstra are easily derived from the LPC cepstra, for example, and the shared state

segmentation ensures that there is minimal increase in computation in the post-feature-extraction stage. The benefit gained here from combining cepstra and Δcepstra feature sets therefore comes at negligible computational expense.

4.7.4 Combining Regular Cepstra with MFCC

MFCC is a perceptually based variation on regular cepstra. MFCC is intended as a *replacement* for standard cepstra, containing information which is more appropriate in terms of *human* perception. It is interesting then to see in Table 4.4 that for the 12 digit sequence the cepstral feature models have a SS EER of 1.93% compared to 2.59% for MFCC. Assuming, as stated previously, that the use of cepstral features in the segmentation model does not create a bias, this indicates that cepstral have more discriminating power. The comparison of the errors made by the two models in Figure 4.13 shows that the errors are mostly complementary, indicating that the feature sets must contain mostly different speaker discriminating information.

In the light of this it is not surprising that the combination of cepstra with MFCC produced substantial improvements in EER with reductions of 21-28% over using cepstra features alone. A summary of the results for the cepstra plus MFCC combination is given in Tables B.19 to B.21.

The combination of Δcepstra with ΔMFCC was less successful with improvements of only 3-18%. It is clear from Table 4.5 that Δ features are best used in combination with static features. A summary of the results for the Δcepstra plus ΔMFCC combination is given in Tables B.28 to B.30.

Note that combining cepstra and MFCC involves considerable extra computation since the feature sets are computed differently. This becomes a key point against this choice of features since the results for the cepstra plus Δcepstra combination are roughly the same as those cepstra plus MFCC, and the former does not involve any significant computational increase.

4.8 State Duration Information

Recall that in the verification score used so far in this chapter the state duration probabilities were omitted, although they *were* used to find the state segmentation in the Viterbi search.

It would seem possible, even likely, that the state duration probabilities contain speaker specific information and that these probabilities could be used as another information stream in

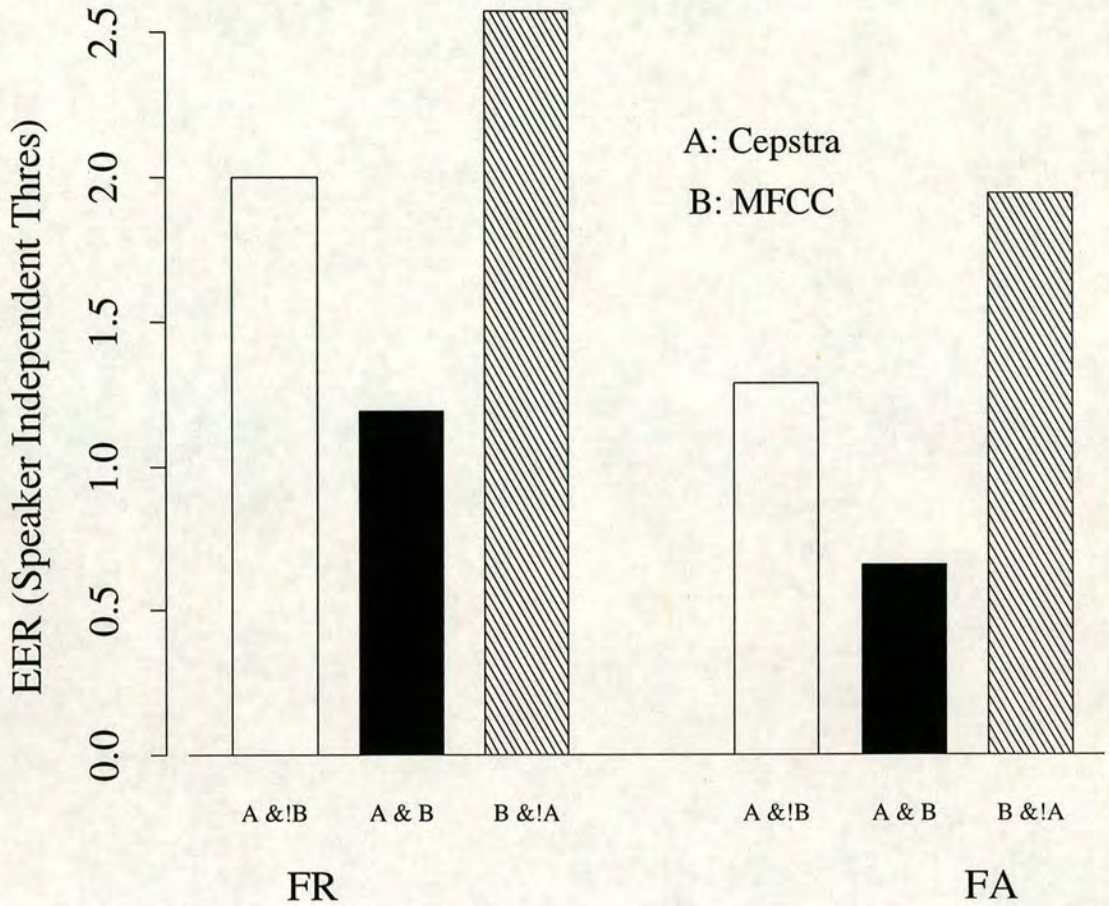

Figure 4.13: Bar-graph showing a comparison of errors created and eliminated by using the MFCC model scores instead of the cepstral scores. The eliminated errors are in white , the created errors in grey and the unchanged errors in black. The two sets of bars represent the FR and FA errors. (SI EER Thresholds).

the verification decision.

The state durations are a by-product of the Viterbi state segmentation so the state duration probabilities can be added to the system at negligible computational cost.

The fact that Φ_{DUR} contains speaker discriminating information is established by using it alone to make the verification decision. The result is an average single digit EER of 24.93% and 20.95% for the SI and SS thresholds respectively. The EER for the 12 digit sequence is compared with the results using single spectral features in Table 4.6. This indicates that duration probabilities *do* contain speaker discriminating information. It would seem intuitively likely that the information from the duration probabilities is complementary to the information from the spectrally-based feature sets. The full summary of results is given in Tables B.13 to B.15.

The next question is how best to use the state duration information. One possibility is to use

Feature	SI Threshold	SS Threshold
	EER	EER
cepstra	3.69	1.93
Δcepstra	9.75	4.4
MFCC	5.08	2.59
State durations	14.57	8.44
ΔMFCC	17.84	9.53

Table 4.6: 12 digit sequence results for 4 different spectral features and the state duration probabilities. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers.

Φ_{TNS} (refer to Equation 3.22) instead of Φ_{OP} as the verification score, but the relative importance of the observation probabilities and duration probabilities is then fixed. A more flexible approach is to treat Φ_{DUR} (defined in Equation 3.24) as another information stream, in the same way as the verification scores from the different feature sets were treated.

1st Feature Set	2nd Feature Set	SI Threshold			SS Threshold		
(α)	($1-\alpha$)	α	EER	Reduct.	α	EER	Reduct.
Durations	cep	0.1	3.57	3%	0.0	1.95	0%
Durations	Δcep	0.4	4.41	55%	0.4	2.54	42%
Durations	MFCC	0.3	4.22	17%	0.3	2.05	21%
Durations	ΔMFCC	0.6	10.47	28%	0.5	5.55	34%
Durations	cep + MFCC	0.0	2.92	0%	0.0	1.45	0%
Durations	cep + Δcep	0.1	2.89	6%	0.1	1.38	1%

Table 4.7: The effect of adding state duration information to the verification decision. Φ_{DUR} is added (using the ratio α) to the single models and pairs of models. All EER are for a 12 digit sequence. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers. *Reduct* is the percentage reduction in error rate gained by adding the state duration information.

The duration scores were combined with each of the 4 feature models and with the two best model pairs. The resulting EERs for the 12 digit sequence are shown in Table 4.7. The duration information is helpful when a single feature model is used, except for the best single feature model (cepstra) which did not benefit from the addition of duration information. The experiments using pairs of models with state duration information show that a weighted linear sum of the duration probabilities with two spectral-feature-based information streams provides no useful benefit.

The summarised results for this experiment are given in Tables B.37 to B.39. This combination of techniques produces the best ASV performance in this chapter, although it must be remembered that a means of estimating the speaker specific digit weights from the training data has not yet been determined.

4.10 Error Analysis

This section looks in more detail at the errors produced by the cepstra plus Δcepstra pair of models on the 12-digit-sequence task using a speaker specific EER. Digit weights are not used in this analysis. Only the *a* block dataset is used in order to ensure complete independence among the test data. Using the *a* block dataset consists of training on the first 5 utterances of each digit and testing on the remaining 20. This gives a total of $21 \times 20 = 420$ true speaker tests and $21 \times 100 = 2100$ impostor tests.

4.10.1 Client Analysis

Figure 4.14 shows a histogram of the the number of clients with various numbers of errors (sum of FR and FA errors), for cepstra plus Δcepstra combination of models. It can be seen that the errors are not uniformly distributed, which indicates that the performance of an ASV system will vary between speakers. There are three clear categories of client performance. Four speakers are *sheep* and have no errors. Three speakers are *goats* and have greater than 10 errors. The middle ground is occupied by the majority of speakers (14) who have between 1 and 5 errors. The worst three clients (the *goats*), who represent 14% of the 21 clients produce 66% of the errors.

4.10.2 Impostor Analysis

Figure 4.15 shows the distribution of FA errors by impostor. Note that there are 101 impostors, since the clients are used as impostors for other clients, but not for themselves. This means that 21 of the impostors have 1/21 fewer trials than the other 80, but this difference is not great enough to affect the general analysis being made here. The trend here is similar to that found for the client speakers. Two of the impostors were successful against 5 different clients, accounting for 14% of the errors. 47 impostors have one or two errors and almost exactly half the impostors

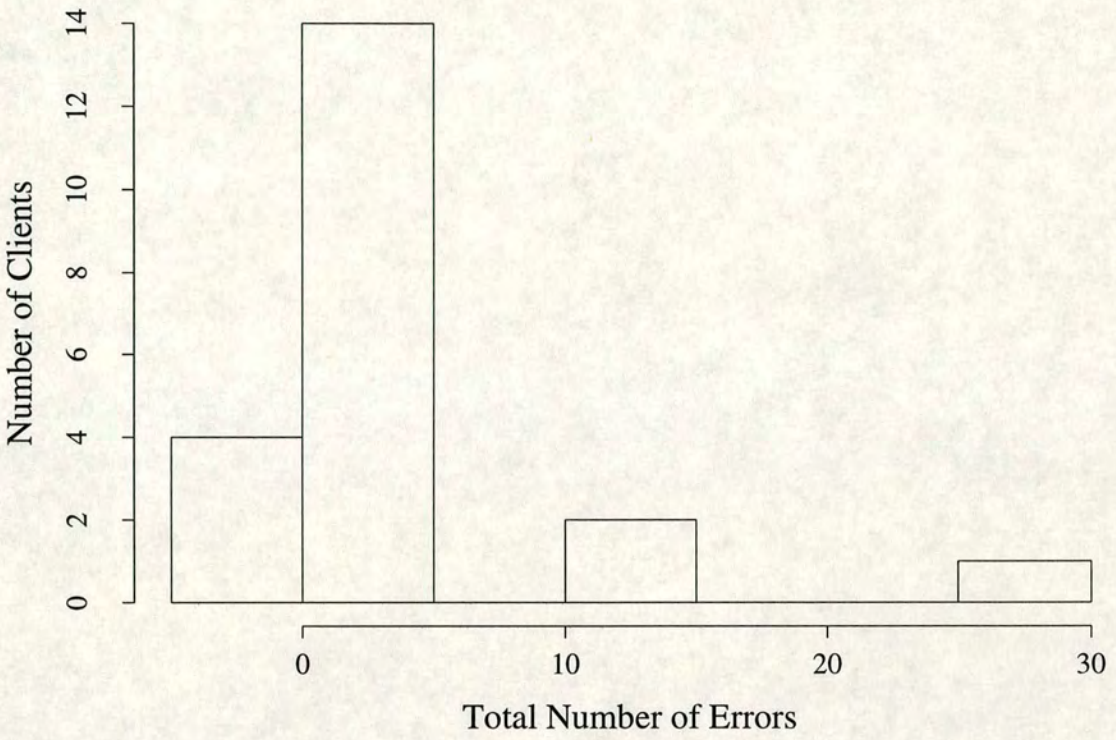

Figure 4.14: Histogram showing the grouping of clients according to the number of errors.

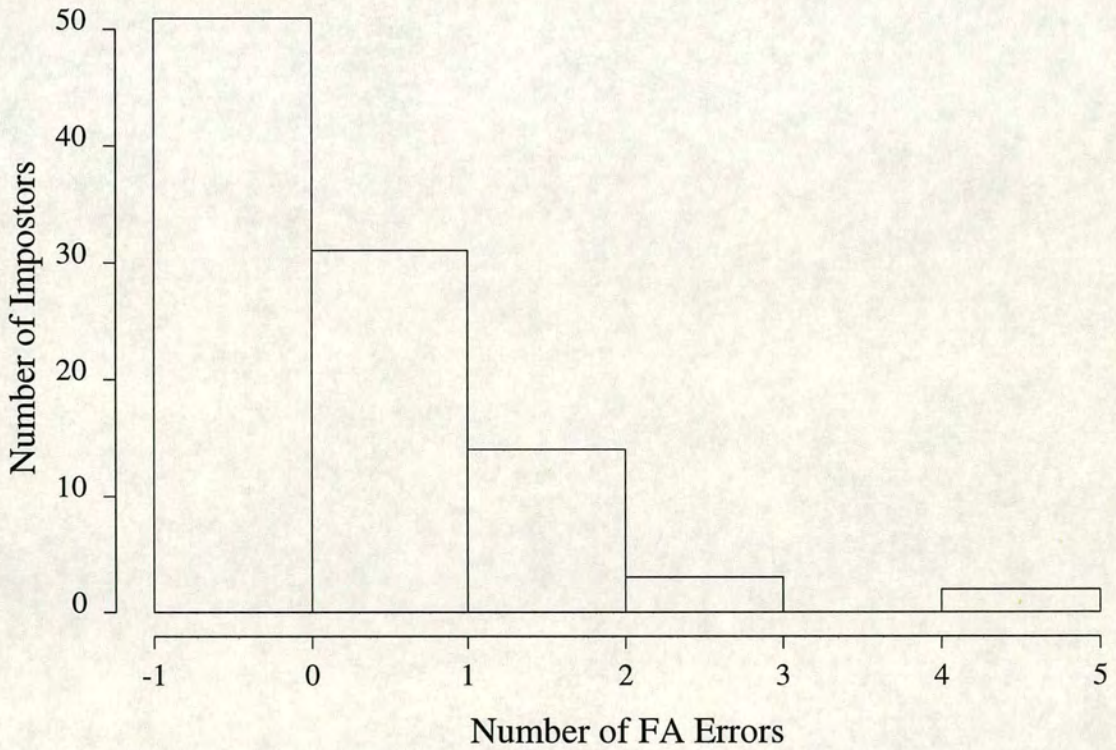

Figure 4.15: Histogram of the Number of Impostors Against False Acceptance Rate. (Speaker specific thresholds, LPC Cepstra)

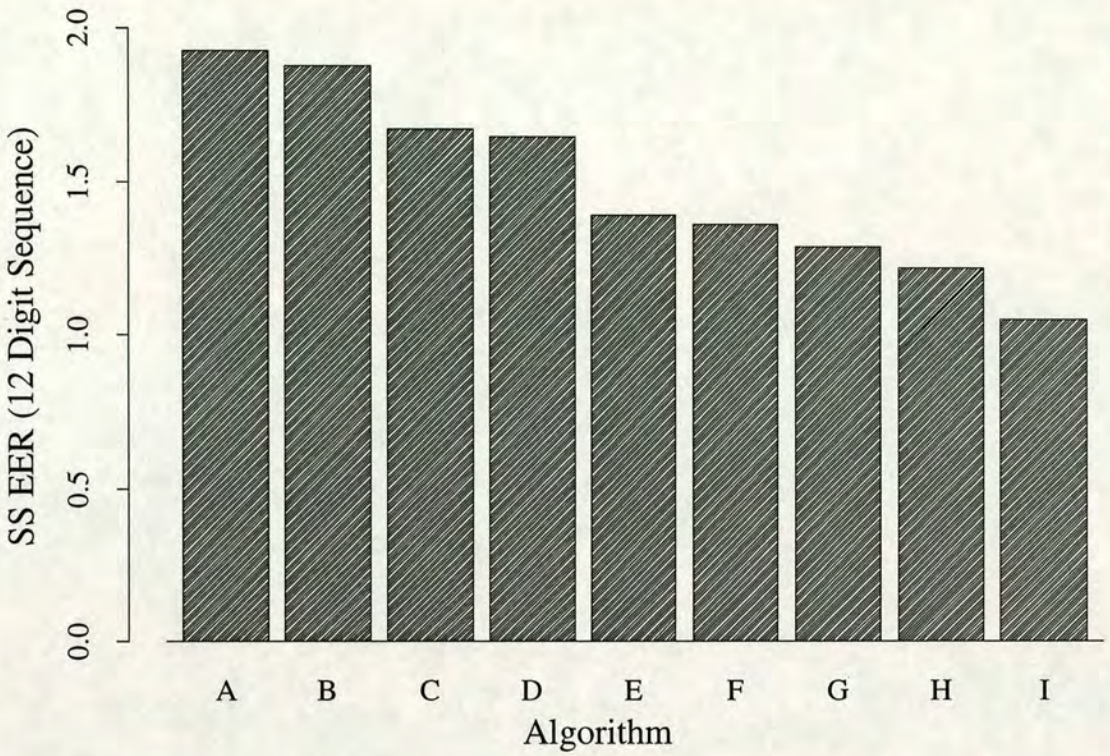

Figure 4.16: Bar plot summarising the various techniques which have produced improvements over the baseline LPC cepstra system. The bars relate the following algorithms. The values are for SS EER on the 12 digit sequence. A: LPC cepstra (baseline system). B: Cepstra system with SI digit weights (c=0.5). C: Cepstra plus ΔMFCC combination. D: ΔCepstra plus MFCC combination. E: Cepstra plus MFCC combination. F: Cepstra plus Δcepstra combination. G: Cepstra system with SS digit weights (c=1). H: Cepstra plus Δcepstra plus MFCC combination. I: Cepstra plus Δcepstra plus MFCC combination with SS digit weights (c=1).

(51) cause no errors.

4.11 Summary

Figure 4.16 gives a summary of the 12-digit sequence SS EER for each of the successful (relative to the baseline) techniques explored in this chapter.

The SI and SS EER results are given in Table 4.8. The percentage improvements are calculated using the single codebook LPC cepstra system as a baseline.

The following conclusions can be made.

- The use of digit weights derived from single digit ASV performance does not provide a useful benefit unless the weights can be specific to the individual speaker. Speaker specific digit weights do provide a reduction in EER even when used with multiple feature models.

Features	SI EER	Reduction	SS EER	Reduction
cepstra (baseline)	3.69	-	1.93	-
SI weights cep. (c=0.5)	-	-	1.88	4%
cepstra + ΔMFCC	3.44	7%	1.67	13%
Δcepstra + MFCC	3.35	9%	1.65	15%
cepstra + MFCC	2.92	21%	1.39	28%
cepstra + Δcepstra	3.08	17%	1.36	30%
SS weights cep. (c=1)	-	-	1.29	33%
cep + Δcep + MFCC	2.52	32%	1.22	37%
cep+Δcep+MFCC SS weights (c=1)	2.29	38%	1.05	46%

Table 4.8: 12 digit sequence results for various techniques used in this chapter. The reduction is the percentage reduction in the EER over the LPC cepstra based baseline system. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers.

- Substantial improvement can be made over the baseline system by adding either a Δcepstra or MFCC feature set to the cepstra feature set in a common state segmentation multiple codebook architecture. The addition of Δcepstra is recommended because it requires minimal additional computation.

- A further slight improvement in performance can be gained by using all three of the above mentioned feature sets, with equal weights. It is likely that some optimisation of the weights would provide a further improvement.

- The use of state duration probabilities does not eliminate any of the more difficult errors when combined, in a linear weighted sum, with the scores from cepstra and Δcepstra features sets.

- The great majority of clients and impostors experience very few errors. Errors are caused by a small number of client speakers being vulnerable to a few different impostors.

The reductions in error rate achieved in this chapter have been substantial, and have not required any unreasonable increase in the computational requirements or complexity of the system. The next chapter will build on the success of the common state segmentation multiple codebook architecture by incorporating a powerful discriminating model into the HASAS framework.

Chapter 5

Discriminative Observation Probabilities (DOP)

This chapter describes a new form of HMM which is designed for improved discrimination in binary classification tasks. Its performance on the ASV task is assessed.

5.1 Motivation for a Discriminative Model

One way to employ HMMs on classification tasks is to model each class with an individual model. The relative likelihoods of the various class models are then used for classification. In speaker identification for example, the classes are the speakers in the identification set and each class is modelled by a speaker dependent model. An utterance is classified as belonging to the speaker whose model has the highest likelihood score for that utterance.

Neural networks are often used as an alternative to HMMs for classification problems. Perhaps the most important advantage in this approach is that a single model with multiple outputs can be used for classification of any number of classes. Not only is it more efficient to use a single model, but more importantly it allows a training algorithm to be used which discriminates between classes.

If the classification task can be reduced to a 2-class or binary decision, the option of using a discriminating model becomes possible with the probability-based HMMs. Assuming that the two classes are mutually exclusive a low likelihood can be used to represent one class and a high likelihood used to represent the other. The likelihoods generated by a discriminating HMM must therefore reflect the likelihood of one class *as opposed to* the other. Several methods of *discriminative training* for HMMs have been proposed, as discussed in Section 2.4.3 -the most

recent being (Liu *et al.*, 1994).

What is proposed here is a method of producing a discriminative model without using discriminative training. The technique of discriminating observation probabilities (DOP) produces a discriminative model by contrasting two standard HMMs. Because no discriminative training is used the DOP model is a model of the *differences between the models* of two classes, rather than the *differences in the classes* themselves.

Discriminative training has met with some success but the disadvantage of the technique is that a commitment must be made to a certain class representation at training time (as defined by the training data chosen to represent the classes).

If, for instance, a model is discriminatively trained with ASV in mind, the training data will be chosen to represent two classes.

1. The client speaker
2. All other speakers

If, on the other hand, the task is closed-set speaker identification, the training data for a given client model would represent two classes

1. The client speaker
2. All other speakers in the identification set (this could be as few as one speaker, and the number could change from day to day).

On a third occasion the task may be that of gender recognition. In this case a discriminative model might be trained with male data versus female data.

The discriminatively trained model is therefore dedicated to a particular discriminative task. A discriminative model constructed from combining standard models has more flexibility in that the models to be combined, and therefore the classes to be discriminated, can be decided at test time rather than at training time.

A second drawback of discriminative training is that the discriminating function is embedded in the training process and is therefore implicit in the discriminative models. DOP models can use any number of different discriminating functions, which can depend on the application, since the discriminating function is defined when the model is constructed at test time.

DOP models therefore aim to exploit the advantages of a discriminating model while avoiding the inflexibility associated with discriminative training.

5.1.1 Rationale for Discriminating Observation Probabilities

The requirement is to construct a discriminative model without using discriminative training. Assume we have a binary classification problem involving two mutually exclusive classes A and B. We also have two models λ_A and λ_B, which model the two classes. The models are not ideal, in the sense that some utterances from class A can score a higher likelihood from λ_B than some utterances from class B and vice versa.

To see how this occurs we need to look at the observation probabilities which are generated within the HMMs. For the j^{th} state of an HMM, an observation probability $b_j(x)$ can be calculated where x is some N dimensional feature vector. If $b_j(x)$ is plotted against x it forms an $N + 1$ dimensional surface.

In order to illustrate this, assume that $N = 1$, such as would be the case if the feature being used was energy, or a single cepstral coefficient.

Figure 5.1 is an illustration of hypothetical 2-dimensional observation probability curves $b_{A,j}(x)$ and $b_{B,j}(x)$ which come from the jth state of two different hypothetical HMMs λ_A and λ_B. On the x-axis is the value of the one-dimensional feature vector x (let's assume its energy).

If we assume that the j^{th} states of the two models represent equivalent acoustic events for the two classes, it is very interesting to compare the two observation probability curves $b_{A,j}(x)$ and $b_{B,j}(x)$.

Some values of energy are more likely for class A than for class B. If these values of energy are observed in the j^{th} state then they are indicators that the utterance is more likely to belong to class A than to class B. Even more importantly, some values of energy are just as likely for class A as for class B, and so *irrespective* of whether the likelihood is high for both or low for both, the occurrence of these values of energy *does not help at all* in classifying the utterance.

It is the values of energy which have similar likelihood for the two classes which would cause errors in classification using a single non-discriminating model. They cause the classification score to be multiplied by probabilities which are independent of the class of the utterance.

What is required in order to construct a discriminating model is to distill the differences in the two models. By comparing the two curves $b_{A,j}(x)$ and $b_{B,j}(x)$ a third curve, a *discriminating observation probability* (DOP) curve $b_{DOP,j}(x)$ can be constructed.

$$b_{DOP,j}(x) = \mathcal{F}_{DOP}(b_{A,j}(x), b_{B,j}(x)) \tag{5.1}$$

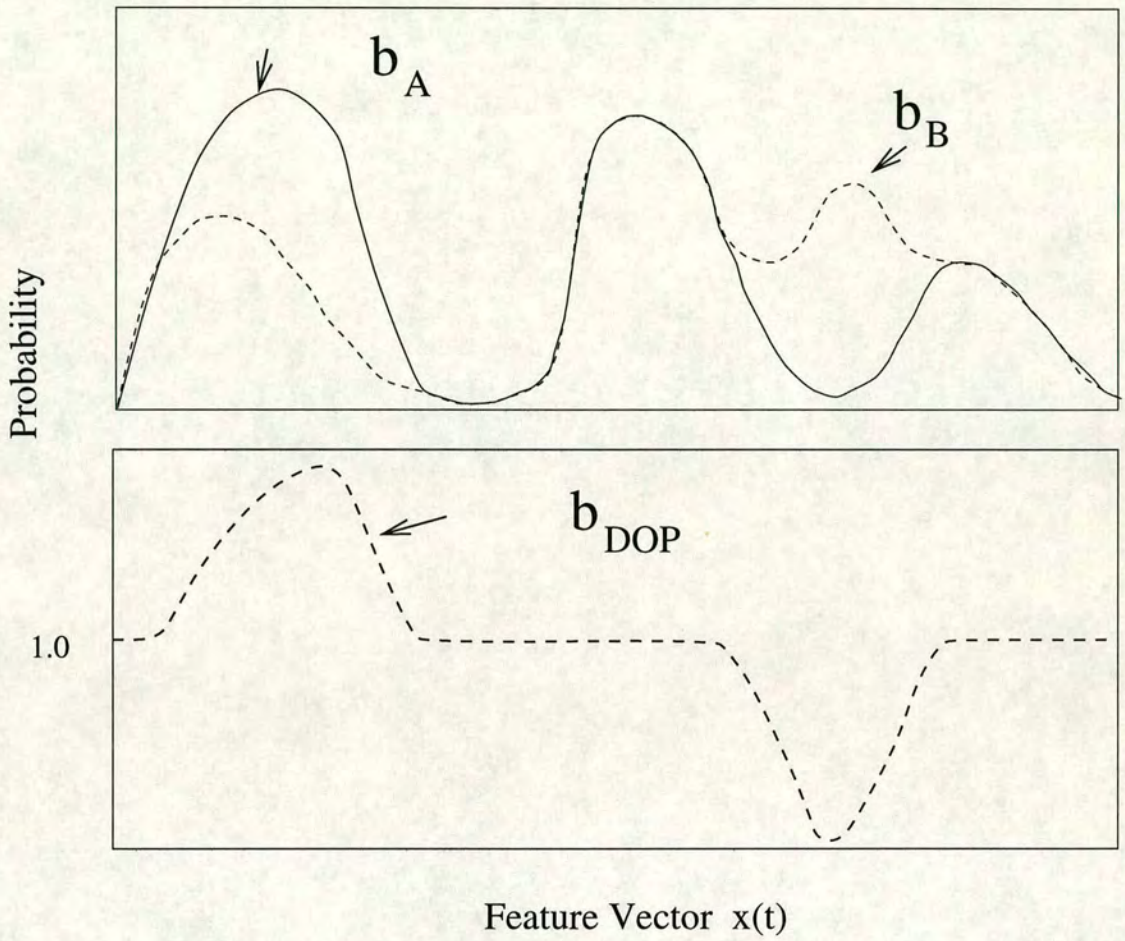

Figure 5.1: One dimensional observation probability surfaces

$\mathcal{F}_{DOP}$ is some discriminating function. In Figure 5.1 a simple ratio is used as the discriminating function $\mathcal{F}_{DOP}$ to calculate $b_{DOP,j}(x)$, as given in Equation 5.2.

$$b_{DOP,j}(x) = b_{A,j}(x)/b_{B,j}(x)) \tag{5.2}$$

Notice that $b_{DOP,j}(x)$ is high for values of energy where class A is more likely than class B, and low when the reverse is true. When the two classes are equally probable, it has a neutral value. For this reason $b_{DOP,j}(x)$ is known as a discriminating observation probability.

It is, in fact, *not* a probability. It is some function of two probabilities as shown in Equation 5.1. In the case of Equation 5.2, $b_{DOP,j}(x)$ is a likelihood ratio. The term observation probability is used because $b_{DOP,j}(x)$ is used *as if it was an observation probability*, as we shall see.

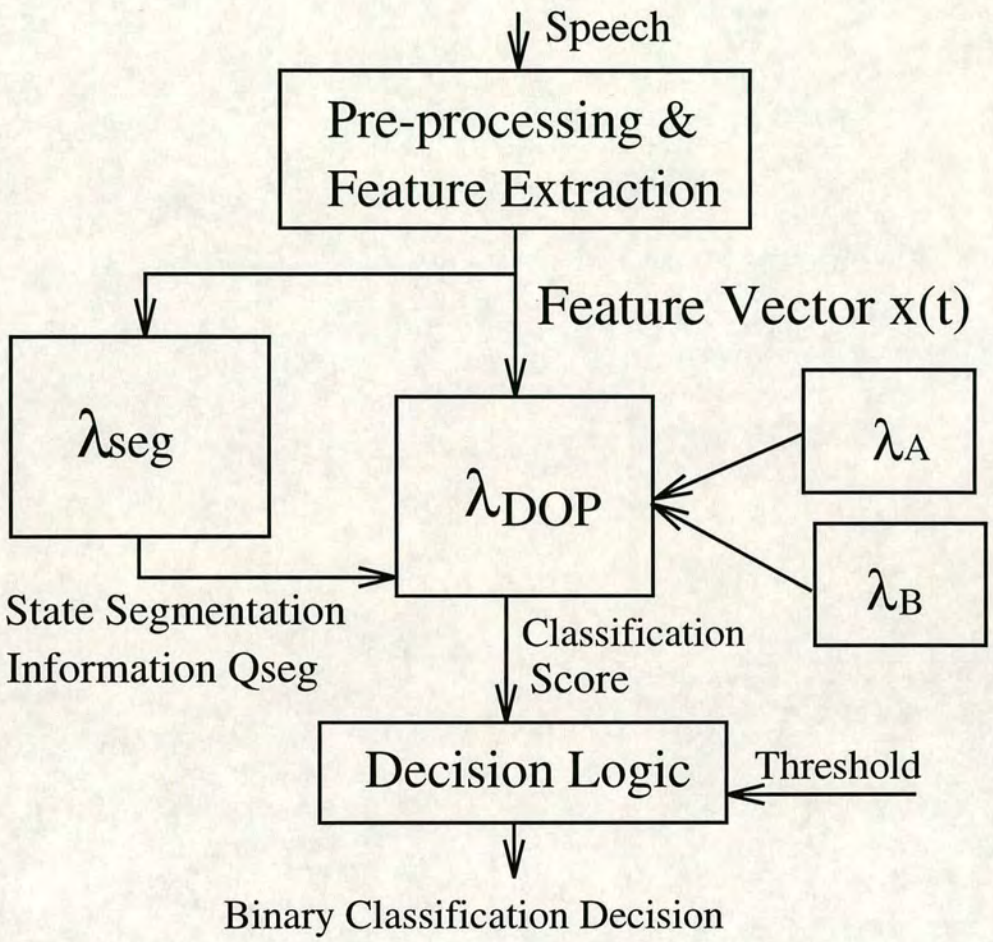

Figure 5.2: Block diagram of the use of a DOP model, constructed by contrasting two class models λ_A and λ_B.

5.2 Constructing a DOP model

The classification of a speech utterance using HMMs consists of four key stages -feature extraction, state segmentation, classification score calculation, and decision logic. The basic structure of a DOP HMM classifier is shown in Figure 5.2.

The use of DOP has the following effects on the four stages of a standard HMM classifier.

- Feature extraction is unchanged, although the relative usefulness of different features may change.
- State Segmentation. In a standard model each class model performs its own state segmentation. Recall the assumption in constructing the DOP model that the j^{th} states in the

two class models represent equivalent acoustic events. This assumption is necessary in justifying the use of a common state segmentation for both class models. The state segmentation model λ_{seg} can be λ_A, λ_B or a third model specifically trained for the purpose. The goal of the state segmentation model is to get consistent, reliable state segmentations for utterances of *either class*. The choice of segmentation model for ASV is examined in Section 5.3.3.

- Classification Score Calculation. The discriminating observation probabilities are calculated according to Equation 5.1. The discriminating observation probability $b_{DOP,j}(x)$ is then used in the back-trace calculation in the usual way. Equations 3.22 and 3.23 take on the form of Equations 5.3 to 5.4. Note that the duration score Φ_{DUR} is not changed. Any of the back-trace techniques discussed in Chapter 3 can be used with DOP models.

$$\Phi_{DOP,TNS} = \frac{\sum_{i=2}^{N-1} \left[\log(d_i(Z_{i+1} - Z_i)) + \sum_{t=Z_i}^{Z_{i+1}-1} \log(b_{DOP,i}(t)) \right]}{\sum_{i=2}^{N-1} (Z_{i+1} - Z_i)} \tag{5.3}$$

$$\Phi_{DOP,OP} = \frac{\sum_{i=2}^{N-1} \sum_{t=Z_i}^{Z_{i+1}-1} \log(b_{DOP,i}(t))}{\sum_{i=2}^{N-1} (Z_{i+1} - Z_i)} \tag{5.4}$$

- Decision logic. The use of DOP models does not restrict the possibilities for decision logic, although it could quite likely influence what the optimal form of decision logic is. Obviously absolute threshold values are affected. DOP scores $\Phi_{DOP,TNS}$ or $\Phi_{DOP,OP}$ are used as information streams in the same way as Φ_{TNS} and Φ_{OP} were in Chapter 4.

5.3 DOP Models For Speaker Verification

Although DOP models can potentially be applied to other binary classification problems, we concern ourselves here with their application to the ASV task. The hypothesis that this chapter examines is that the discriminating observation probability surface can be used in the verification stage to improve verification performance. It is an intuitively appealing concept since ASV is a straight-forward binary classification problem. The results in this chapter will show that the theory works very well in practice.

Automatic speaker verification is a classification task in which the two classes are the *client speaker* and the *impostors* or the set of *all other speakers*.

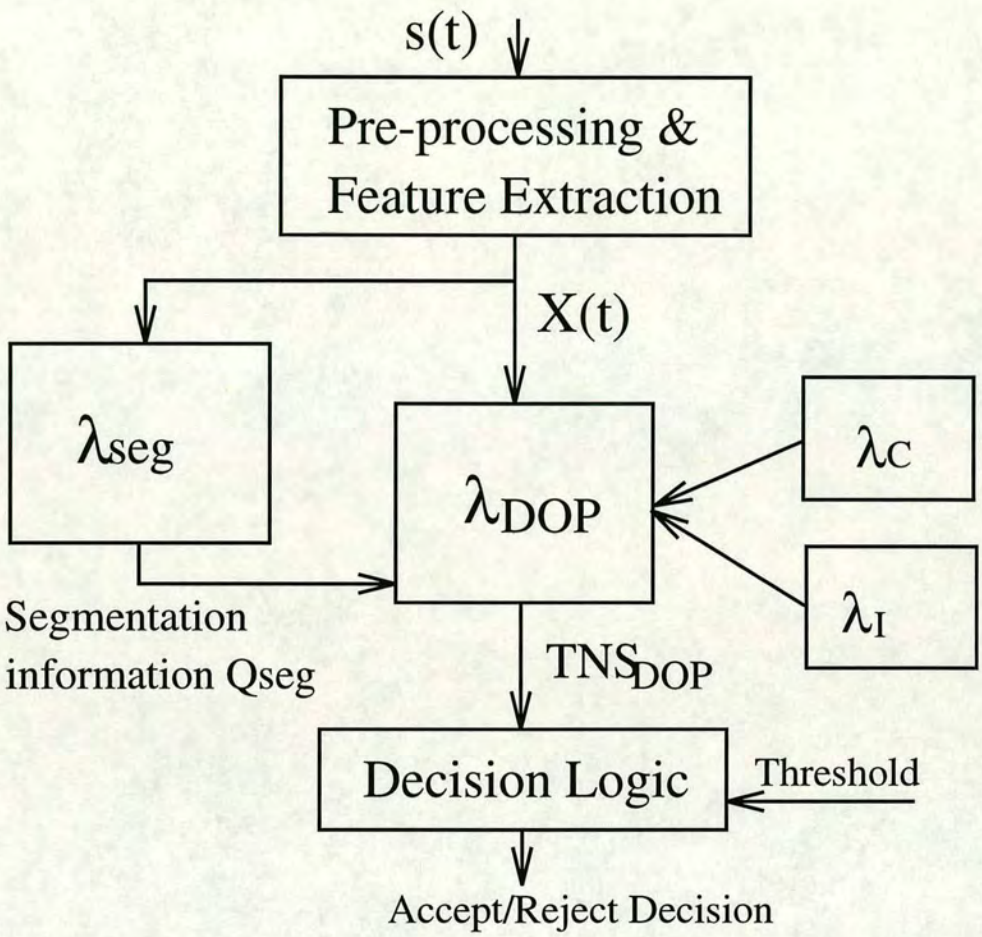

Figure 5.3: Block diagram of the use of a DOP model, constructed by contrasting a client model λ_C and an impostor model λ_I.

Figure 5.3 is a modified version of Figure 5.2 showing a block diagram of the way DOP models can be used for ASV.

5.3.1 Choosing an Impostor Model

The impostor model (λ_I) can be constructed in a variety of ways. The criteria for choosing an impostor model are similar to those used to select an impostor model for speaker normalisation. Some of the methods which have been proposed for the construction of a impostor model for systems using speaker normalisation for ASV are discussed in detail in Section 2.4.4. All of the impostor models discussed, including cohort speakers, can be used as impostor models in the DOP architecture. Although it is not yet clear what the best choice of impostor model is, the

speaker independent model has computational advantages, and appears to be a sensible choice. A speaker independent impostor model is used throughout this chapter. The speaker independent models are trained using data from 80 speakers, as described in Section 3.4.6.

5.3.2 Constructing the Client Model

A speaker dependent model of the client speaker can be used as the client model (λ_C). Recall the earlier assumption that the j^{th} state of the two class models (the client and impostor models in this case) should correspond to equivalent acoustic events. This equivalence is encouraged during the training procedure by using the speaker independent word models (the impostor models) to initialise or *seed* the training of the speaker dependent client models. This could be taken a stage further in future work by using a one-pass Viterbi training for the client models based on a Viterbi state segmentation using the impostor models.

5.3.3 Constructing the Segmentation Model

There are three ways to perform the state segmentation.

1. DOP-C. DOP models using the state segmentation from the speaker dependent (SD) client model. ($\lambda_{seg} = \lambda_C$)
2. DOP-I. DOP models using the state segmentation from the speaker independent (SI) impostor model. ($\lambda_{seg} = \lambda_I$)
3. DOP-IND. DOP models using the state segmentation from a third, independent, model which is specifically constructed for the segmentation task.

The DOP-C and DOP-I approaches are likely to be useful for ASV and are investigated and compared in Section 5.6.

The use of an independent λ_{seg} is likely to be useful in other classification tasks such as gender recognition where the following models might be used.

- λ_A= male speakers
- λ_B= female speakers
- λ_{seg}= speaker independent

5.4 Single Information Stream

This section details a series of experiments which parallel those of Section 4.1. The performance of DOP models is evaluated using models based on each of the four feature sets. The performance of the DOP models can then be assessed in comparison with the use of conventional models.

In this section, and throughout this chapter, the verification probability used is $\phi_{DOP,OP}$ (the product of the DOP along the Viterbi path), as given in Equation 5.4. No duration probabilities are included.

Table 5.1 shows the 12-digit-sequence EER for models using each of the four feature sets, compared to the conventional model result from Section 4.1. The percentage reduction in EER from using the DOP model instead of the conventional model is given in each case. For all feature sets the use of a DOP model improved performance substantially. The reductions in EER ranged from 43% to 90%.

	SI EER			SS EER		
Feature	Con	DOP	Reduct	Con	DOP	Reduct
cepstra	3.69	2.12	43%	1.93	0.79	59%
Δcepstra	9.75	1.88	81%	4.40	0.5	89%
MFCC	5.08	2.81	45%	2.59	1.22	53%
ΔMFCC	17.84	4.57	74%	9.53	0.98	90%

Table 5.1: Single feature set results. SS and SI EERs are given for the 12-digit-sequence. Reduct refers to the reduction in EER from using DOP instead of conventional models (Con).

The results summaries for each of the features sets are in Tables C.1 to C.12.

The 12 digit sequence EER is reduced 43-59% for the cepstra and MFCC features and 74-90% for the Δ features. DOP Δcepstra has the lowest EER and the ΔMFCC results improved substantially from clearly the worst feature set in the conventional models to being comparable with the other three features for the DOP models. This indicates that the DOP technique is particularly effective with delta features, although it is not clear why this is the case.

It is interesting that the different feature sets have considerably different relative performance levels for the different clients. An examination of the breakdown of errors by client for the 12-digit-sequence in Tables C.3, C.6, C.9 and C.12 is condensed in Table 5.2.

As well as having the lowest average EER the Δcepstra-based model is error-free for 13/21 (62%) of the clients. The *best* column shows that if the best feature model for each client

Speaker	best	worst	mean	cep	Δcep	MFCC	ΔMFCC
1	0	0.8	0.2	0	0	0.8	0
2	0	0.3	0.1	0	0	0.3	0
3	0	1.5	0.5	0.3	0	0.3	1.5
4	0	3.5	1.2	0.7	0	3.5	0.7
5	0	1.3	0.4	1.3	0	0	0.3
6	0	0.7	0.3	0.4	0	0.7	0.2
7	0	1.8	0.8	0	0	1.8	1.4
8	0	7.0	2.7	0.7	0	7.0	3.1
9	0.3	0.6	0.4	0.4	0.4	0.3	0.6
10	0	0	0	0	0	0	0
11	0	3.5	2.0	1.5	2.8	3.5	0
12	0	0	0	0	0	0	0
13	0	0.3	0.1	0.2	0	0	0.3
14	0	0.7	0.5	0.7	0	0.1	1.0
15	0	0.1	0	0	0.1	0	0
16	0.1	3.7	1.3	3.7	0.2	0.1	1.0
17	0	0.6	0.3	0.1	0.6	0	0.6
18	0	0	0	0	0	0	0
19	1.3	4.3	3.1	4.3	1.3	5.0	1.7
20	2.2	4.9	3.6	2.2	5	2.2	4.9
21	0	3.3	0.9	0.1	0.1	0	3.3
mean	0.2	1.85	0.8	0.8	0.5	1.2	1.0

Table 5.2: Individual feature set results by client speaker. 12 digit sequence SS EER. The last four columns are the breakdown of errors by client for the four different feature models. The column labelled *best* contains the best EER over all the feature models for each client speaker. The *mean* column has the average EER over the 4 feature models for each of the clients. The final row contains the average over each column.

could be known *a priori*, then only 4 (19%) of the client speakers would have any errors. This is an indication of considerable independence between the feature sets, and close examination of Table 5.2 reveals many more examples of such independence. Client 8 had no errors with the LPC Δcepstra models but 7% EER using the MFCC models. Client 19 also had a large EER using the MFCC feature set but a much smaller error when the Δcepstra features are used. Client 20, on the other hand shows the reverse, with a preference MFCC over for Δcepstra. Clearly different feature sets emphasise different aspects of the speech signal and these vary in importance from client to client.

One way to capitalise on this would be to find some way to determine *a priori* which features are most useful for each client in terms of intra versus inter-speaker variability and use that

feature set in the DOP models for that client. It may be quite feasible to do this by doing a closed test on the training data using each of the feature sets.

An alternative approach is the multiple information stream approach introduced in Chapter 4. This approach has the advantage that several feature sets can be used together to make a more robust decision. It is also a more general approach since stream weights can be made speaker specific if required, and the degenerate case of this using binary weights is the same as selecting the best model or models. Using non-binary weights allows the system to take advantage of any independence in the speaker discriminating information of the different feature sets. The use of multiple information streams based on DOP models is investigated in the next section.

5.5 Pair-wise Combinations of Information Streams

This section shows the results of combining the information from various pairs of models. The relative weighting of the two information streams is determined by the same *a posteriori* experimental approach used in Chapter 4. Two different types of model combination suggest themselves. The first possibility is that, although the DOP models are clearly superior to the conventional models, the two types of model may be usefully combined. This approach is investigated in Section 5.5.1. The other possibility is to combine DOP models based on different feature sets. Section 5.5.2 compares the performance of DOP model pairs with single DOP models and Section 5.5.3 compares the performance of DOP model pairs with that of conventional model pairs.

1st Feature Set	2nd Feature Set	α		EER			
		SI	SS	SI	Reduct	SS	Reduct
DOP cepstra	cepstra	0.8	1.0	2.01	5%	0.79	0%
DOP MFCC	MFCC	0.8	0.9	2.01	28%	0.80	34%
DOP cep +DOP Δcep	cep + Δcep	0.8	0.9	1.36	3%	0.39	1%
DOP cep + DOP MFCC	cep + MFCC	0.9	0.9	1.53	3%	0.52	4%

Table 5.3: Pair-wise combinations of DOP models with conventional models. EER for 12 digit sequence. The value of α gives the weighting of the first information stream, with weighting $1 - \alpha$ for the second. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers. *Reduct* is the percentage reduction in error rate gained by using the pair instead of using the better model on its own.

5.5.1 Combining DOP Models with Conventional Models

It can be seen in Table 5.3 that the conventional cepstra and DOP cepstra models are not very complementary, since using a conventional model and a DOP model based on the same feature set provides very little reduction in the EER. The exception to this is the MFCC feature set. The conventional and DOP MFCC models can be combined to produce reductions in SS and SI EER of 28 and 34% over the use of the DOP MFCC model alone. It should be noted however that the pair of MFCC models are no better than a single DOP cepstra model, so this result is not an argument for using a DOP-plus-conventional combination.

Further experiments were performed in order to determine whether conventional models could improve performance in a system with four information streams. The conventional cepstra and Δ cepstra models were combined with DOP cepstra and DOP Δ cepstra models to obtain four information streams for the verification decision. The results are given in Table 5.3.

The relative weightings between models of the same type (conventional/DOP) were taken from the optimum values of α determined in earlier experiments. The relative weighting between the conventional pair and DOP model pair was determined by trying all values from 0 to 1 in steps of 0.1 in the usual way. The value of α quoted in the table is therefore the relative weighting between the conventional pair and DOP model pair.

For example, the weightings for the SI EER of the second to last row of Table 5.3 are given in Equation 5.5.

$$\alpha \times (0.2 \times \Phi_{\mathrm{DOP,cep}} + 0.8 \times \Phi_{\mathrm{DOP,\Delta cep}}) + (1 - \alpha)(0.7 \times \Phi_{\mathrm{cep}} + 0.3 \times \Phi_{\mathrm{\Delta cep}}) \qquad (5.5)$$
$$\alpha = 0.8$$

A further four-model experiment was performed using the conventional and DOP cepstra plus MFCC combinations. The four model results show consistent but very slight improvement from the use of conventional models with DOP models. The fact that the improvement is so small indicates that the DOP models should *replace* rather than *complement* the conventional models.

This conclusion contrasts with the findings of earlier experiments (Forsyth & Jack, 1994; Forsyth *et al.*, 1994) in which the DOP models were only useful as a complement to conventional

models. The relative lack of success of DOP models in those earlier experiments was due to using a poorly trained impostor model (only 20 speakers were used).

5.5.2 Combining Multiple DOP Models

The combination of two information streams from models based on two different feature sets produced considerable benefits in Chapter 4. The apparent independence of the DOP models for the four parameter sets discussed in Section 5.4 is a sign that improvements in performance could also be gained by combining the scores from two DOP models.

1st Feature Set	2nd Feature Set	α		EER			
		SI	SS	SI	Reduct	SS	Reduct
DOP cepstra	DOP Δcepstra	0.2	0.3	1.10	42%	0.21	58%
DOP cepstra	DOP MFCC	0.7	0.7	1.48	30%	0.49	38%
DOP cepstra	DOP ΔMFCC	0.2	0.2	1.23	42%	0.17	78%
DOP Δcepstra	DOP MFCC	0.8	0.9	1.12	41%	0.25	50%
DOP Δcepstra	DOP ΔMFCC	0.8	0.4	1.64	13%	0.26	49%
DOP MFCC	DOP ΔMFCC	0.2	0.1	1.60	43%	0.38	62%

Table 5.4: Pair-wise combinations of DOP models based on different feature sets. EER for 12 digit sequence. The value of α gives the weighting of the first information stream, with weighting $1 - \alpha$ for the second. SS means speaker specific thresholds were used to calculate the EER, SI means that the thresholds for the EER were the same for all speakers. *Reduct* is the percentage reduction in error rate of using both models together rather than using the better model on its own.

There are six possible pair-wise combinations of models and Table 5.4 contains a summary of six sets of experiments. For each pair of information streams the result given is for the optimum value of information stream weight (α). The reduction in 12-digit-sequence EER quoted is that gained by using the pair of DOP models instead of the better DOP model on its own.

The improvements are very substantial, and are even greater than the improvements gained by combining two conventional models. Combining two conventional models produced reductions in 12-digit-sequence EER from 4-36%. Combining two DOP models produces improvements from 13-78%.

As would be expected, the combination of cepstra with MFCC is not as effective as the combination of either of these with a Δ feature set. This is because cepstra and MFCC are similar feature sets. The best combination for the SS EER, by a very narrow margin, is cepstra

with ΔMFCC. This could be because this combination has maximum independence between feature sets - a static LPC cepstra feature set with a dynamic MFCC feature set.

Taking into account both the SI EER and SS EER results, the best combination is cepstra plus Δcepstra. It is possible, however, that the use of a cepstra based state segmentation is providing a slight bias in favour of the cepstra parameter set.

Combining More Than Two Information Streams

The combination of any two DOP models produces a reduction in EER of 13-78%. This indicates independence in the information contained in any two feature sets and so a combination of all the feature sets should produce even further reductions in EER.[1].

Model Combination	SI		SS	
	EER	TDM	EER	TDM
cepstra/Δcepstra	1.10	4.79	0.21	5.42
All four DOP models	0.96	4.77	0.26	5.46

Table 5.5: Comparison of the cepstra/Δcepstra DOP model combination with the combination of all four DOP models (cepstra, Δcepstra, MFCC, ΔMFCC). The performance measures are the EER and the Targeted Distance Measure (TDM) for the 12-digit-sequence using both SI and SS EER thresholds.

An explorative experiment was performed combining all four DOP models with equal weighting but, as Table 5.5 shows, it was inconclusive. The SI EER is reduced but the SS EER is increased. The targeted distance measure (TDM) introduced in Section 2.2.3 was used to determine if there was any increase in separation of the critical areas of the client and impostor score distributions. The values are very similar for the two algorithms, making it unlikely that there is much difference between them.

Of course the weights for the four information streams have not been optimised. If they were it would guarantee performance at least as good as any pair of information streams. This experiment indicates, however, that even with optimal weighting, the combination of more than two of the DOP models is unlikely to produce a large increase in performance -at least on this database.

[1] Assuming, once again, that a way can be found to determine appropriate weights.

In practice, because the Δ features are very computationally cheap to derive from the original feature set, the feature set combinations most likely to be used are the static-dynamic combinations of cepstra/Δcepstra and MFCC/ΔMFCC. Combining this fact with the experimental results, cepstra plus Δcepstra is recommended as a useful DOP model combination.

5.5.3 DOP Pairs versus Conventional Pairs

Model Pair	SI EER			SS EER		
	CON	DOP	Reduct	CON	DOP	Reduct
cepstra/Δcepstra	3.08	1.10	64%	1.36	0.21	85%
cepstra/MFCC	2.92	1.48	49%	1.39	0.49	65%
cepstra/ΔMFCC	3.44	1.23	64%	1.67	0.17	90%
Δcepstra/MFCC	3.35	1.12	67%	1.65	0.25	85%
Δcepstra/ΔMFCC	9.38	1.64	83%	4.25	0.26	94%
MFCC/ΔMFCC	4.37	1.60	63%	2.22	0.38	83%

Table 5.6: Comparison of DOP versus conventional for several pair-wise combinations. EER for 12 digit string. The percentages are the percentage reduction in EER obtained by using the DOP models instead of the conventional models (CON).

Section 5.5.2 compared the use of two DOP models with the use of a single DOP model. This section compares the use of a pair of DOP models with the use of the equivalent pair of conventional models. Table 5.6 gives the comparison for the six pair-wise combinations. The evidence favouring DOP models is clear, with reductions in 12-digit-sequence EER in the range of 49-94%.

5.6 Choosing a Segmentation Model

Section 5.3.3 suggested several possible choices of segmentation model. The segmentation model used so far has been the speaker dependent client model ($\lambda_{seg} = \lambda_C$), but this is not the only logical choice. The argument for using the client model is that it is a speaker dependent model and is likely to produce a good segmentation because it is specific to the client's speech. It is well known that speaker dependent models provide better speech recognition than speaker independent models.

The argument for using the speaker independent models for state segmentation is that they are trained using 17 times more data than the speaker dependent models and are therefore likely

to be more robust, especially in the case of impostor speech.

	SI EER			SS EER		
Feature Model	$\lambda_{seg} = \lambda_I$	$\lambda_{seg} = \lambda_C$	Reduct	$\lambda_{seg} = \lambda_I$	$\lambda_{seg} = \lambda_C$	Reduct
cepstra	2.14	2.12	11%	0.89	0.79	11%
Δcepstra	2.56	1.88	27%	0.63	0.50	21%
MFCC	2.92	2.81	4%	1.24	1.22	2%
ΔMFCC	5.43	4.57	16%	1.25	0.98	22%

Table 5.7: Single model results. EER are for the 12 digit sequence. The percentages are the percentage reduction in EER obtained by using the λ_C instead of λ_I.

This section examines the difference between using the two different segmentation models. Table 5.7 shows the comparison for a single DOP model. There is a small but consistent advantage in favour of the speaker specific segmentation model ($\lambda_{seg} = \lambda_C$). Of more practical interest, however, is which segmentation model performs best when a pair of DOP models is used, particularly the favoured cepstra plus Δcepstra combination.

Model Pair	SI EER			SS EER		
	$\lambda_{seg} = \lambda_I$	$\lambda_{seg} = \lambda_C$	Reduct	$\lambda_{seg} = \lambda_I$	$\lambda_{seg} = \lambda_C$	Reduct
cepstra/Δcepstra	1.55	1.10	29%	0.52	0.21	60%
cepstra/MFCC	1.50	1.23	18%	0.39	0.17	56%
cepstra/ΔMFCC	1.61	1.48	8%	0.40	0.49	-22%
Δcepstra/MFCC	1.29	1.12	13%	0.44	0.25	43%
Δcepstra/ΔMFCC	2.31	1.64	29%	0.47	0.26	45%
MFCC/ΔMFCC	1.72	1.60	7%	0.60	0.38	37%

Table 5.8: Comparison of two state segmentation models (λ_{seg}) for several pair-wise combinations. EER are for the 12 digit sequence. The percentages are the percentage reduction in EER obtained by using the λ_C instead of λ_I.

Table 5.8 condenses the results of a further series of experiments using pairs of DOP models. The differences between the results for the two segmentation models are greater than they were when using a single model, but the trend is not completely consistent. The results using the client model for state segmentation are 7-60% better than those using the impostor model, except for the cepstra plus ΔMFCC model combination, which produced a lower SS EER when using the impostor model as a segmentation model. Note, however, that the best performing model pairs, namely cepstra/Δcepstra, cepstra/MFCC and Δcepstra/MFCC all indicate a preference for the client model as a segmentation model.

5.7 Assessing Bias in the Reference Model

Ideally the reference model should be constructed from a single token of as many speakers as possible (preferably several hundred) and none of those speakers should be included in the set of client speakers or the set of impostor speakers. This would best reflect the *real-world* conditions of a speaker verification system. In commercial applications speaker independent models would be constructed by the company producing the system and these reference models would then be used for various applications, which would involve completely independent sets of speakers.

Due to the limited size of the database it was not possible to have complete independence between the impostor set and the set of speakers used to train the impostor model (the reference speakers).

The speaker independent impostor models for these experiments were trained using a single token of each digit from 80 speakers. None of the client speakers were included in this set, thus ensuring independence between the set of client speakers and the set of reference speakers. However, 69 of the 80 reference speakers were included in the impostor set, so there is only partial independence between the impostor set and the set of reference speakers. Also the *same utterances* from these 69 speakers which were used to train the reference model were used as impostor data. This raises the question of whether the reference model biased the system against the data from those 69 speakers.

It is unlikely that any bias would be great because, although the reference model may have seen an impostor utterance in its training, that utterance was only one of 80 training utterances and so the model is not going to be strongly *tuned* to that impostor. It was considered important, nevertheless, to experimentally determine whether a strong bias was present.

In order to do this, experiments were run using a smaller impostor set of 23 speakers that did not include the 69 speakers who were used for the reference model. These 23 speakers were in fact the client speaker set, with the claimed speaker left out, plus two other speakers.

Table 5.9 shows that although the error rates did increase with the smaller impostor set, the fact that the increase was similar whether or not the reference model was used indicates that this is due to the 23 speaker impostor set being more difficult than the 100 impostor set and not to any reference model bias.

The percentage improvement gained from the use of DOP is similar for the 23 and 100 impostor sets. On average the improvements using the 23 impostor set were 4 percentage points

lower than the improvements for the 100 impostor set. This could indicate a slight experimental bias but since the DOP models still show great improvement using the 23 impostor set, this bias will not affect our conclusions in any way.

It should also be remembered that 23 impostors is too small for a very reliable estimate of error rates. This experiment was performed solely to determine whether the reference models were biased against the 100 impostor set.

	SI EER			SS EER		
Feature	23 CON	23 DOP	Reduct (100)	23 CON	23 DOP	Reduct (100)
cepstra	5.01	2.61	48%(43%)	2.18	1.02	53% (59%)
Δcepstra	10.50	2.31	78%(81%)	4.95	0.98	80% (89%)
MFCC	5.36	3.08	43%(45%)	3.46	1.82	47% (53%)
ΔMFCC	17.16	5.29	69%(74%)	10.71	1.57	85% (90%)
cepstra,Δcepstra	3.81	1.31	66%(58%)	1.74	0.36	79% (85%)
cepstra, MFCC	3.39	1.64	52%(58%)	1.60	0.60	63% (88%)
cepstra, ΔMFCC	4.54	1.73	62%(57%)	2.09	0.33	84% (71%)
Δcepstra, MFCC	3.46	1.51	56%(67%)	2.02	0.43	79% (85%)
Δcepstra, ΔMFCC	9.76	2.02	79%(83%)	4.69	0.55	88% (94%)
MFCC, ΔMFCC	4.64	2.19	53%(63%)	2.68	0.68	75% (83%)

Table 5.9: Comparison of DOP versus conventional (CON) EER using a completely independent 23 impostor set and the semi-independent 100 impostor set. All error rates are for the 23 impostor set, the improvement for the 100 impostor set is in parenthesis for comparison. Although the improvements using DOP were generally slightly less when the 23 impostor set was used, they were not very different. SS means speaker specific thresholds were used to calculate the EER and SI means the same threshold was used for all speakers. EERs are for the 12 digit sequence, using single and paired feature models.

5.7.1 Stability of α values

The results for the 23 speaker set provide a good opportunity to determine in a very rough way how invariant the values of α are between test data sets. Note, however, that the test sets are only partially independent. The client data is the same and 23% of the impostor data is the same.

Table 5.10 shows the optimal values of α for each pair of feature models for the 23 impostor and 100 impostor data sets.

The α values show little variation between 23 and 100 impostor sets. It is hoped that the use of a common state segmentation across the different feature models means that the values of

	CON				DOP			
	SI EER		SS EER		SI EER		SS EER	
Features	23	100	23	100	23	100	23	100
cepstra,Δcepstra	.5	.7	.5	.6	.3	.2	.3	.3
cepstra, MFCC	.7	.9	.7	.7	.7	.7	.8	.7
cepstra, ΔMFCC	.6	.9	.7	.7	.5	.7	.2	.2
Δcepstra, MFCC	.6	.7	.7	.8	.8	.8	.9	.9
Δcepstra, ΔMFCC	.7	.8	.7	.7	.6	.8	.5	.4
MFCC, ΔMFCC	.6	.6	.4	.5	.2	.2	.1	.1

Table 5.10: Comparison of α values for SS and SI EER thresholds, and model types. DOP denotes the use of DOP models and CON denotes the use of conventional models. The numbers 23 and 100 refer to the number of speakers in the impostor set. All values of α were found by minimising the EER for a 12 digit sequence.

α are reflective of the relative usefulness of the different feature sets for the *speaker* modelling part of the verification process. Although these results show α to be fairly stable, there is only partial independence between the 100 and 23 impostor experiments. To find out if values of α exist which are universally optimal comparative experiments on another system using different training and test data and different model topologies would be needed.

5.8 Comparing DOP With Speaker Normalisation

5.8.1 Introduction

The range of techniques known collectively as *speaker normalisation* were introduced in Section 2.4.4. In this section speaker normalisation will be examined from the viewpoint of the DOP framework. It will be seen that although the techniques are different, they can have some similarities. The techniques are moves in the same direction and speaker normalisation can be viewed as a first step towards DOP. By taking a special case of DOP, a more direct comparison with speaker normalisation can be made. The results will show that speaker normalisation only captures part of the power of the DOP approach.

5.8.2 Framework for Comparing Speaker Normalisation and DOP

Speaker verification is a classification problem. Test utterances must be classified into *client* and *impostor* classes. There are several ways of achieving this.

1. Client model.

   The traditional approach to speaker verification with HMM is to use speaker dependent word or phone models to model the speaker, as was done in Chapter 4. The difficulty with this approach is that only the client class is being modelled. If the match with the client speaker model is not *good enough* the test utterance is assumed to belong to the impostor class. This places a heavy emphasis on the accuracy of the client model. Any way in which the test utterance does not fit the client model is taken to be an indication that the utterance came from an impostor. Any inadequacies in the client model will therefore increase the probability of false rejection. Given the limited training data generally available in speaker verification applications, the assumption of a detailed and robustly trained client model is not realistic. The use of a single, speaker dependent model alone can be referred to as a *client-model* (CM) approach.

2. Client and Imposter Models. If a second model, a model of the impostor class, is used in addition to the client model, a comparison can be made to determine which class the test utterance should be assigned to. This can be considered a *client and impostor model* (CIM) approach. Speaker normalisation is an example of this approach. Figure 5.4 is a block diagram of the use of models of both the client and impostor classes. λ_C is the speaker specific model and λ_I is the impostor model.

3. Discriminative Models. A third possibility is the use of a discriminating model, either a discriminatively trained model or a DOP model.

The aim of this section is to examine the similarities between the DOP and CIM approaches. The two models are therefore configured to be as directly comparable as possible. In particular, the following choices have been made.

1. The same SI impostor model is used for DOP and CIM.

2. A log likelihood ratio will be used as the DOP function $\mathcal{F}_{DOP}$, as given in Equation 5.2.

3. The verification score normally used with speaker normalisation is ϕ_{TNS} (refer to Equation 3.22). In this study ϕ_{OP} will also be used in the CIM approach. The verification score ϕ_{CIM} is therefore defined as follows.
   $\phi_{CIM} = \phi_{C,OP} - \phi_{I,OP}$.

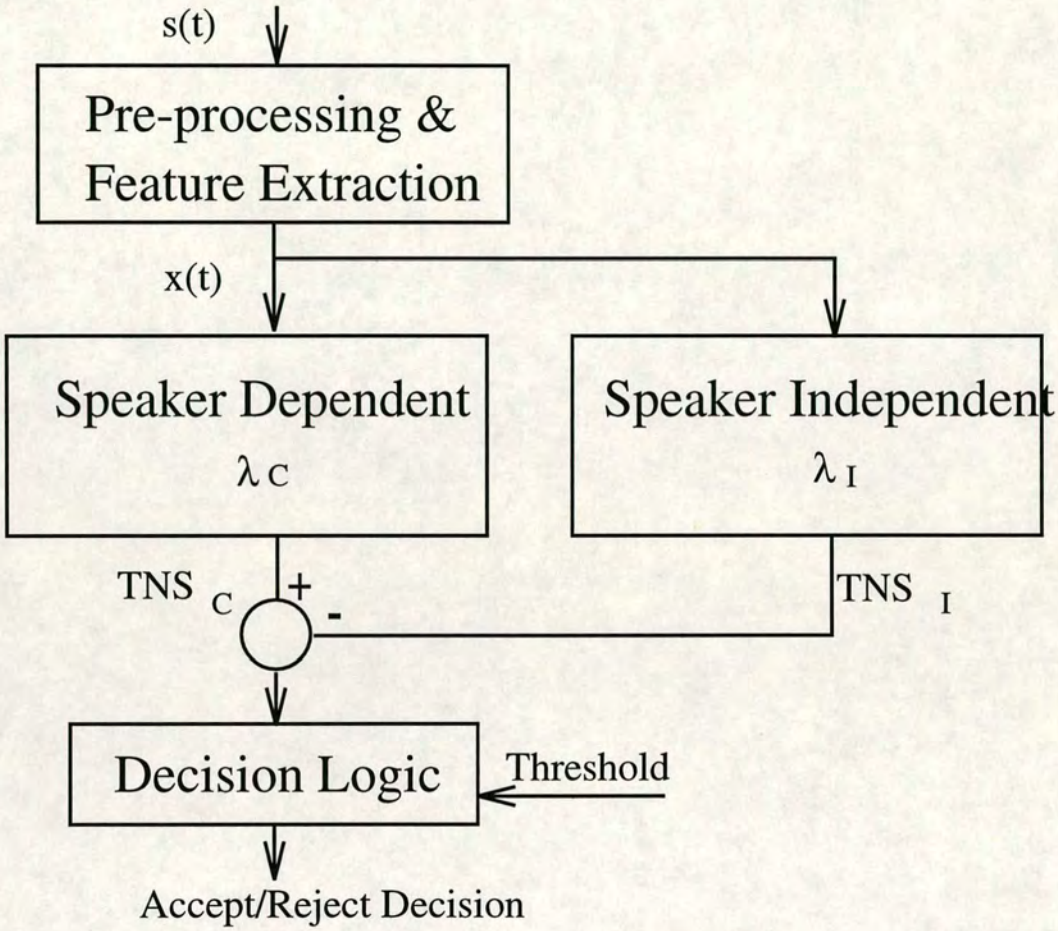

Figure 5.4: Block diagram of the CIM approach to ASV which uses two class models, a client model λ_C and an impostor model λ_I, which can be a speaker independent model or a group of cohort speaker models.

It is a debatable point whether this this last item is the correct approach, after all ϕ_{OP} is not generally used for speaker normalisation[2]. The point is, however, that it *could* be used in the CIM approach, and so it does not represent a fundamental difference between the speaker normalisation and DOP approaches. In a similar manner cohort speaker models are used more often than speaker independent impostor models for speaker normalisation, but cohort speaker models can also be used to construct DOP models, so that is not a fundamental difference between the two approaches and is not a reason for preferring one approach ahead of another.

[2] An exception to this is (Rosenberg *et al.*, 1992) in which ϕ_{OP} is used with a separate word duration constraint.

The fundamental difference between the DOP and CIM architectures is that the use of a single state segmentation for the DOP models ensures that the client and impostor observation probabilities which are being compared in constructing the DOP models are related to the same events. With the CIM architecture the client and impostor models each perform their own state segmentation. Wherever the client state segmentation differs from the impostor model state segmentation ($Q_C(t) \neq Q_I(t)$), the observation probabilities from the two models relate to different speech events. Because $Q_{seg}(t)$ is used to calculate both the client and the impostor observation probabilities (b_C and b_I) in the DOP model, their comparison is always meaningful[3]. The use of a common state segmentation makes frame by frame comparisons of observation probabilities more valid, because like is always being compared with like.

Note that the validity of frame by frame comparisons makes it possible to use more sophisticated forms of discriminating function ($\mathcal{F}_{DOP}$) than a simple log likelihood ratio, as is discussed in Section 5.9.

Another way of viewing the two approaches is that the DOP technique addresses the fundamentals of discrimination within a single model rather than comparing two models outwith the modelling process. For this reason it is expected that the performance of DOP models will be superior to a CIM approach.

Tables 5.11 and 5.12 show a comparison between the DOP and speaker normalisation approaches, using single models and pairs of models.

	SI EER			SS EER		
Feature	CIM	DOP	Reduct	CIM	DOP	Reduct
cepstra	2.17	2.12	2%	0.98	0.79	19%
Δcepstra	2.63	1.88	29%	0.81	0.50	38%
MFCC	3.24	2.81	13%	1.63	1.22	25%
ΔMFCC	6.31	4.57	28%	2.05	0.98	52%

Table 5.11: Comparison of DOP HMM with the CIM (speaker normalisation) approach, using a single information stream. EERs are for the 12-digit-sequence.

The DOP HMMs produce the same or better results as the normalisation technique in all cases. Using a single information stream the DOP models are superior by 2-52% and using two information streams the improvement was 3-55%.

[3] Assuming that the same states in λ_C, λ_I and λ_{seg} correspond to the same acoustic events.

	SI EER			SS EER		
Feature	CIM	DOP	Reduct	CIM	DOP	Reduct
cepstra, Δcepstra	1.19	1.10	8%	0.34	0.21	38%
cepstra, MFCC	1.67	1.48	11%	0.69	0.49	29%
cepstra, ΔMFCC	1.65	1.23	25%	0.36	0.17	53%
Δcepstra, MFCC	1.16	1.12	3%	0.40	0.25	38%
Δcepstra, ΔMFCC	2.43	1.64	33%	0.58	0.26	55%
MFCC, ΔMFCC	2.17	1.60	26%	0.84	0.38	55%

Table 5.12: Comparison of DOP HMM with the CIM (speaker normalisation) approach, using two information streams. EERs are for the 12-digit-sequence.

A: DOP cepstra + Δcepstra					B: CIM cepstra +Δcepstra
Significance Level	FR Num	FR Rate	n_{10}	n_{01}	n_{11}(common errors)
0.013	3	0.71%	2	12	5

Table 5.13: Comparison between algorithms A and B for the 12-digit-sequence using SS thresholds. Significance level is the probability of sampling numbers of FA errors at least as different as n_{10} and n_{01} if the performance A and B is equivalent at the specified FR rate. This test is on the *a* block dataset only.

The cepstra plus Δcepstra case is of particular interest because it is the favoured combination. Using speaker specific thresholds, the DOP models had a better TDM (5.41) than the CIM models (5.24). Table 5.13 gives the significance level of the difference between the DOP and CIM architectures for the combination of cepstra and Δcepstra[4]. The probability of so large a difference in the results if the DOP and CIM architectures have equivalent performance is 1.3%. It is reasonable to conclude from these results that the DOP HMMs should be used in preference to speaker normalisation (CIM).

Note that the DOP models have been configured to be as much like the CIM approach as possible. DOP models are potentially much more powerful. In particular, the log likelihood ratio discriminating function used is very simple. Speaker normalisation uses the difference in the time-normalised sum of the log observation probabilities whereas the DOP-based verification score is a time-normalised sum of the differences in the log observation probabilities, which is very similar (the only difference is in the state segmentation). Other discriminating functions could take more advantage of the frame by frame discrimination which is made possible by the

[4] Refer to Section A for details of the significance test.

DOP models. This is the subject of the next section.

5.9 Optimising the Discriminating Function $\mathcal{F}_{\mathrm{DOP}}$

As stated previously, the discriminating function $\mathcal{F}_{\mathrm{DOP}}$ does not need to be a simple log likelihood ratio, although this has proven to be a successful choice. $\mathcal{F}_{\mathrm{DOP}}$ can be a more sophisticated (possibly non-linear) function, and can be state or model specific if required. Finding an optimal form for $\mathcal{F}_{\mathrm{DOP}}$ will not be an easy task, but it is likely to be rewarding. This section looks at what is going on inside the DOP models on a frame by frame basis, in order to get a better idea of how the discriminating function might be optimised.

In general the difference in the observation probabilities of the client and impostor models ($\mathrm{b_C(t)/b_I(t)}$) will be a measure of how much the feature vector $\mathrm{x(t)}$, for frame t, is like the client speaker as opposed to any other speaker. An exception to this can, however, be imagined. Take the situation where $\mathrm{b_I(t)}$ is very low. Since a speaker independent model is being used as the impostor model, a low value of $\mathrm{b_I(t)}$ represents a low *speech recognition* score for that frame. One explanation for this is that the wrong speech has been uttered. This is always a possibility, and a commercially deployed text-dependent ASV system would have to have a speech recogniser as a first pass to check this. Assume that this has been done and that the utterance is correct. The low value of $\mathrm{b_I(t)}$ must then be due either to an inadequacy in the impostor model, or to some form of noise in the utterance which has not been observed in the large amount of data, from a range of speakers, which was used to train the impostor model. In either case the frame concerned is of little use in the classification task being performed, since it is not well modelled. An ideal discriminating function would cope with such frames, perhaps by assigning them a neutral value for $\mathrm{b_{DOP}(t)}$, or perhaps by deleting them from the back-trace calculation by using the frame-weighting scheme described in Section 3.4.9. A realistic measure of what $\mathrm{b_I(t)}$ is *too low* would have to be determined.

Another argument against $\mathcal{F}_{\mathrm{DOP}}$ being a simple ratio is that it does not distinguish between cases where $\mathrm{b_C(t)}$ and $\mathrm{b_I(t)}$ are both high and cases where they are both low. Once again we can use the argument that low values in $\mathrm{b_I(t)}$ can be due to inadequacies in modelling. If both $\mathrm{b_C(t)}$ and $\mathrm{b_I(t)}$ are low then that is an indication that the feature vector $\mathrm{x(t)}$ is not well modelled, for whatever reason.

For example take the case of a frame t where $\mathrm{b_C(t)}$ and $\mathrm{b_I(t)}$ are both very low. Now

assume that because λ_I has been trained with much more data than λ_C it is a little more robust to unexpected values of $x(t)$ and $b_I(t) = 2 \times b_C(t)$. Now imagine a second frame t' where both $b_C(t')$ and $b_I(t')$ are high and $b_C(t') = 1.5 \times b_I(t')$. In this case the feature vector is well modelled by both models and the frame is clearly an indication that the utterance came from the client.

If $\mathcal{F}_{DOP}$ is a simple ratio, then for these two frames we get $b_{DOP}(t) = 0.5$ and $b_{DOP}(t') = 1.5$. The average log of these two values is -0.06 which, if only these two frames were used, would lead to the utterance being falsely rejected (DOP model thresholds are likely to be around zero). Clearly differences in the observation probabilities when both probabilities are high should have more importance than similar differences when they are both low. An ideal $\mathcal{F}_{DOP}$ would reflect this.

There are other modifications to $\mathcal{F}_{DOP}$ which could be helpful. For example, a text-independent system using log likelihood ratios for speaker normalisation, placed a limit on the magnitude of the log likelihood ratio on the grounds that robustness would be increased if extreme values were eliminated (Bahler *et al.*, 1994). This approach could also be applied in text-dependent tasks.

Much of this reasoning is intuitive and is not based on experimental evidence, but it does nonetheless suggest that a better discriminating function could be found.

The frame scores for a single FR error were plotted to see if any of these potential deficiencies in $\mathcal{F}_{DOP}$ could actually be observed. Figure 5.5 shows the frame scores from the client, impostor and DOP models for the digit *eight*. The areas highlighted with circles are areas where both the client and the impostor score are low. It can be seen that these cases produce misleading values in the DOP model. This digit was part of a 12 digit string which produced one of the three errors on the *a block* dataset using DOP cepstra + DOP Δcepstra models. Examination of plots for the other digits reveal several other instances of low probabilities producing misleading DOP scores. An examination of this single 12-digit-sequence suggests that setting a minimum log probability for the client and impostor scores of -4, and ignoring all frames where both the client and the impostor scores are below -3 might be beneficial. This corresponds to a $\mathcal{F}_{DOP}$ which hard limits its inputs, combined with binary frame weights. If further errors of both types (FR and FA) were studied, it is possible a good $\mathcal{F}_{DOP}$ could be developed. However, the construction of $\mathcal{F}_{DOP}$ using rules derived from experimental data is likely to be very time consuming and to

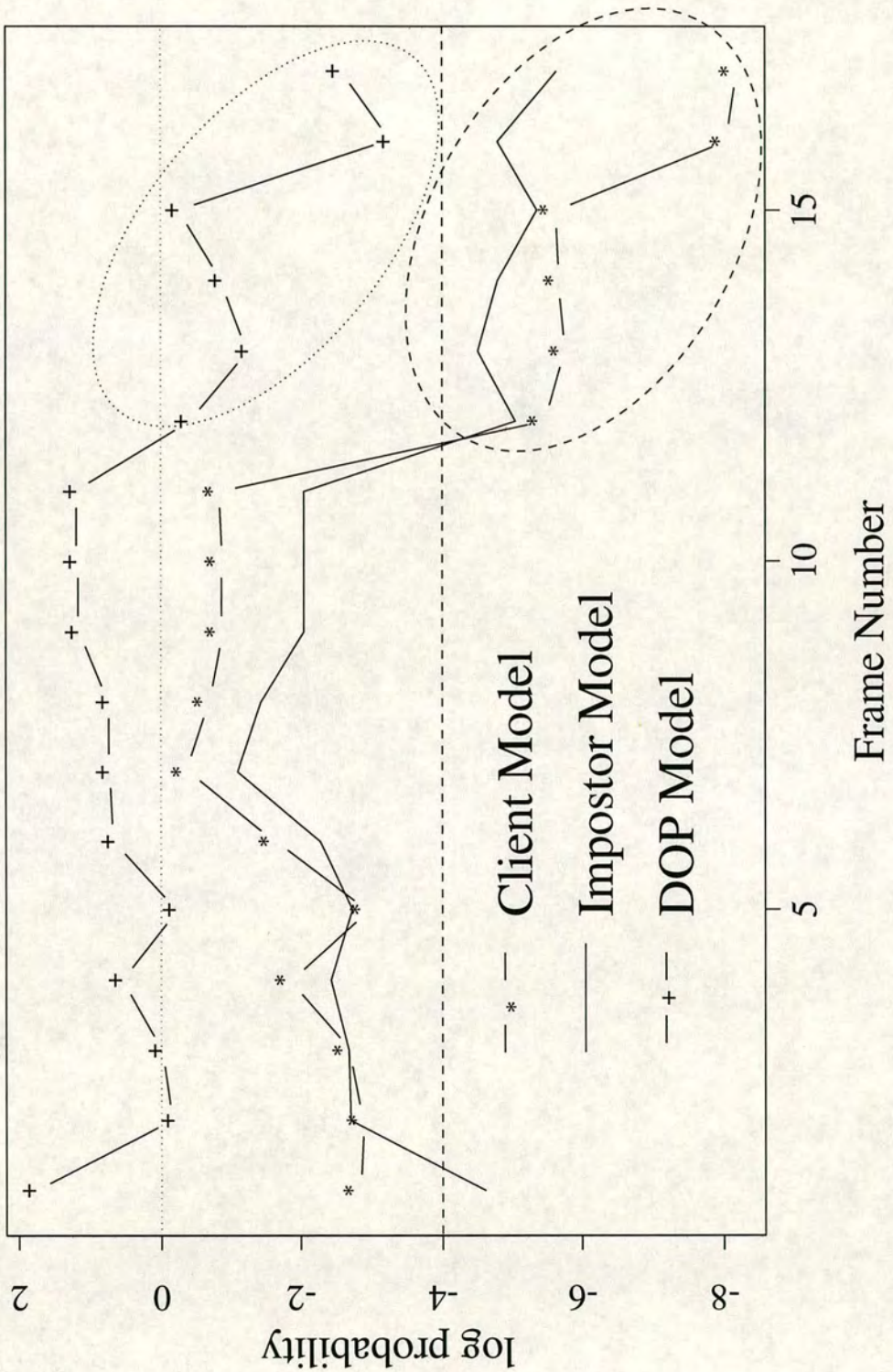

Figure 5.5: Frame scores for client, impostor and DOP models for the digit eight. This digit is taken from a 12-digit sequence which caused a FR error. The DOP values should ideally average above zero. The areas where low client and impostor probabilities are causing misleading DOP scores are highlighted.

miss several subtle effects.

Training a neural network to perform the discriminating function would be an alternative approach. If the neural net is initialised to perform a log likelihood ratio then it can only improve performance (provided representative data is used to train it). The observation probabilities from different feature sets could all be combined at the frame level in the same neural net, thereby providing a means of optimally combining more than two feature sets.

The exploration of possible forms for $\mathcal{F}_{\mathrm{DOP}}$ is recommended as an exciting area for further research.

5.10 Analysis of Errors

This section looks in more detail at the errors which remain when using the combination of DOP cepstra and DOP Δcepstra models. Only the *a* block dataset is used in order to ensure the complete independence among the test data. Using the *a* block dataset consists of training on the first 5 utterances of each digit and testing on the remaining 20. This gives a total of $21 \times 20 = 420$ true speaker tests and $21 \times 100 = 2100$ impostor tests.

Speaker specific thresholds are used throughout this analysis because it is felt that errors caused by a sub-optimal threshold are in a different category to errors caused by a failure of the models to distinguish client and impostor utterances. It is the latter type of error that the DOP models aim to reduce, and using speaker specific thresholds allows us to focus on these errors.

Figure 5.6 shows the breakdown of errors by client and by impostor. 94 of the 101 impostors causes no errors. Of the remaining seven impostors, six caused one error each and the seventh only two errors. This indicates that there are no particularly successful impostors.

The histogram of errors by client speaker, which considers FA and FR errors is also shown in Figure 5.6. It shows that errors have been eliminated for 17 out of 21 (81%) of clients for the 12-digit-sequence. If this database is reflective of the actual user population then this is an excellent level of initial performance for the system, considering the small amount of enrolment data that is required. The total number of FR errors was 3/420 (0.71%) and the total number of FA errors was 7/2100 (0.3%). These percentages are different because of quantisation effects[5]. In order to assess the statistical significance of these error rates we assume that both the FR and

[5]The number of true speaker tests is small (n=420) giving a quantisation in the FR rate of +/-0.24%. The quantisation of the FA rate is 0.05% so equal error rates are not usually possible.

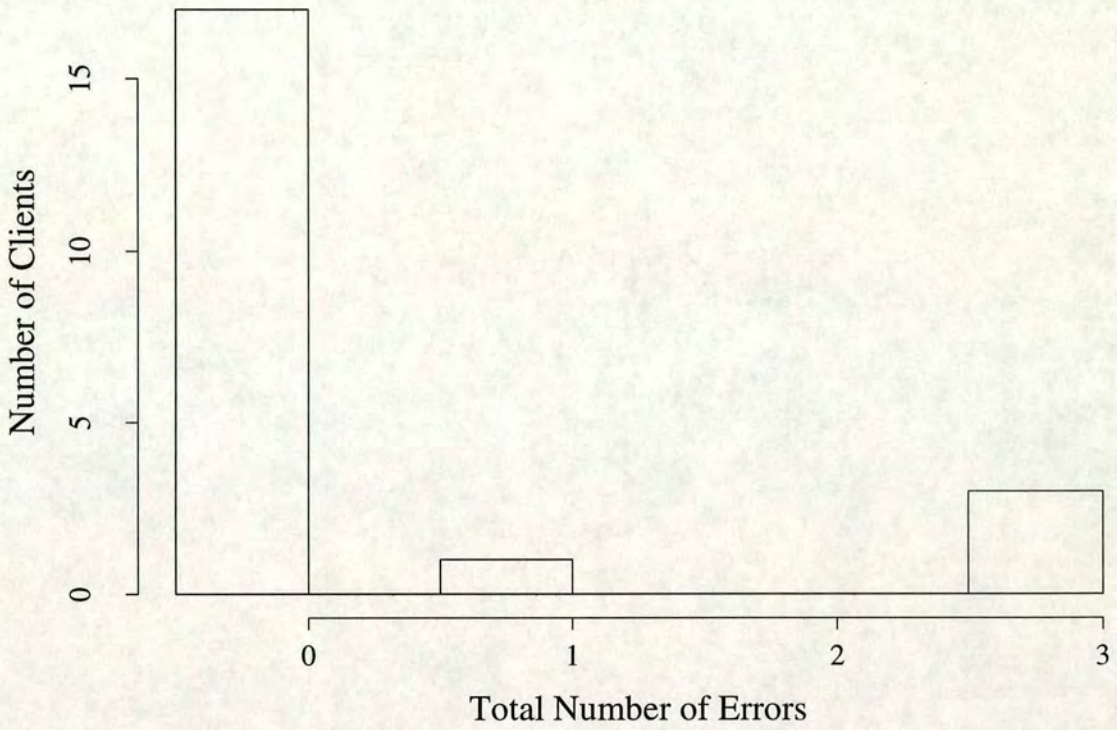

Figure 5.6: Break-down of errors by client and impostor for the DOP cepstra plus Δcepstra model combination. A SS EER is used on the 12-digit-sequence for the *a* dataset.

the FA errors are binomially distributed[6] and apply hypothesis tests to the null hypotheses that the FR/FA rates are less than or equal to 1%. The probability of obtaining less than or equal to k errors is given by Equation 5.7. The resulting values of significance level α are 0.4 (FR) and 0.0004 (FA). Note, however, that absolute error rates are of limited interest because they are so task-dependent.

$$\alpha = \sum_{x=0}^{k} P(R = x|n, \pi) \tag{5.6}$$

$$k = 3(FR), 7(FA) \quad n = 420(FR), 2100(FA)$$

The ZFR rate is 5%, which is arguably a reasonable error rate for a commercial system. This is important since it is likely that usability will be more important than security in a telephone banking/shopping type environment, and that very low FR rates will be required. The ZFA rate is 19% which shows that very high security is still not achievable with small amounts of training data and a 12-digit-sequence test utterance. It is unlikely that any application exists

[6] Refer to Equation A.1

where security is a high enough priority for a 20% FR rate to be tolerated. If increasing the amount of enrolment and test data does not improve the ZFA rate sufficiently then high security applications would need to employ ASV only as a complement to other security measures.

Absolute values of performance are of little interest however because all the sources of variation between ASV tasks discussed in Chapter 2 mean that comparison of error rates between systems is almost meaningless. Unless the task definition *exactly* matches the intended application, any performance predictions are likely to be very unreliable. What *is* meaningful is the comparison between algorithms using the same system, and the advantage of the DOP cepstra + DOP Δcepstra pair of models over the baseline cepstra model is beyond doubt, as shown in Table 5.14[7]. If the thresholds of the two algorithms are set so that each algorithm produces 3 FR errors, the DOP algorithm produces no FA errors that are not also produced by the baseline system. The baseline system, however, produces 213 FA errors which are not produced by the DOP models. This difference is significant at the $\log(\alpha) = -148$ level, which makes α zero for all practical purposes.

A: DOP cepstra + Δcepstra					B: baseline cepstra
Significance Level	FR Num	FR Rate	n_{10}	n_{01}	n_{11}(common errors)
0.0	3	0.71%	0	213	7

Table 5.14: Comparison between algorithms A and B for the 12-digit-sequence using SS thresholds. Significance level is the probability of sampling numbers of FA errors at least as different as n_{10} and n_{01} if the performance A and B is equivalent at the specified FR rate. This test is on the *a* block dataset only.

Figure 5.7 gives a comparison of the EER curves for the two algorithms and it is clear that apart from a reduction in the EER, there has been a noticeable decrease in the critical tails of the client and impostor score distributions, resulting in big decreases in ZFR and ZFA rates (76% and 80% respectively). Figure 5.8 compares the receiver operating characteristic of the two algorithms which gives an idea of the differences in performance at various operating points.

5.11 Summary

[7] Refer to Section A for details of this significance test.

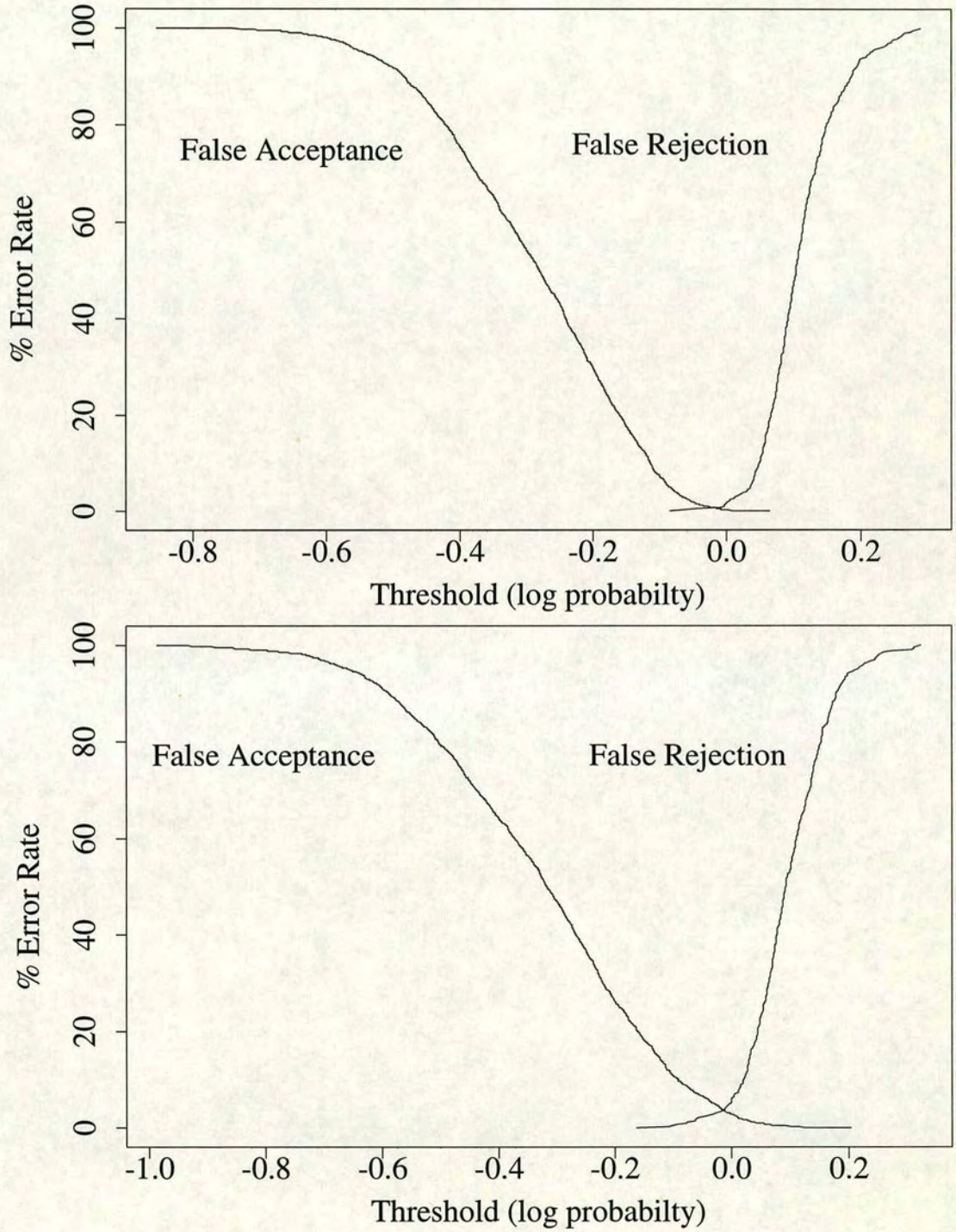

Figure 5.7: Equal error rate curves for the DOP cepstra plus Δcepstra model combination (top) and the baseline cepstra models (bottom). A SS EER is used on the 12-digit-sequence.

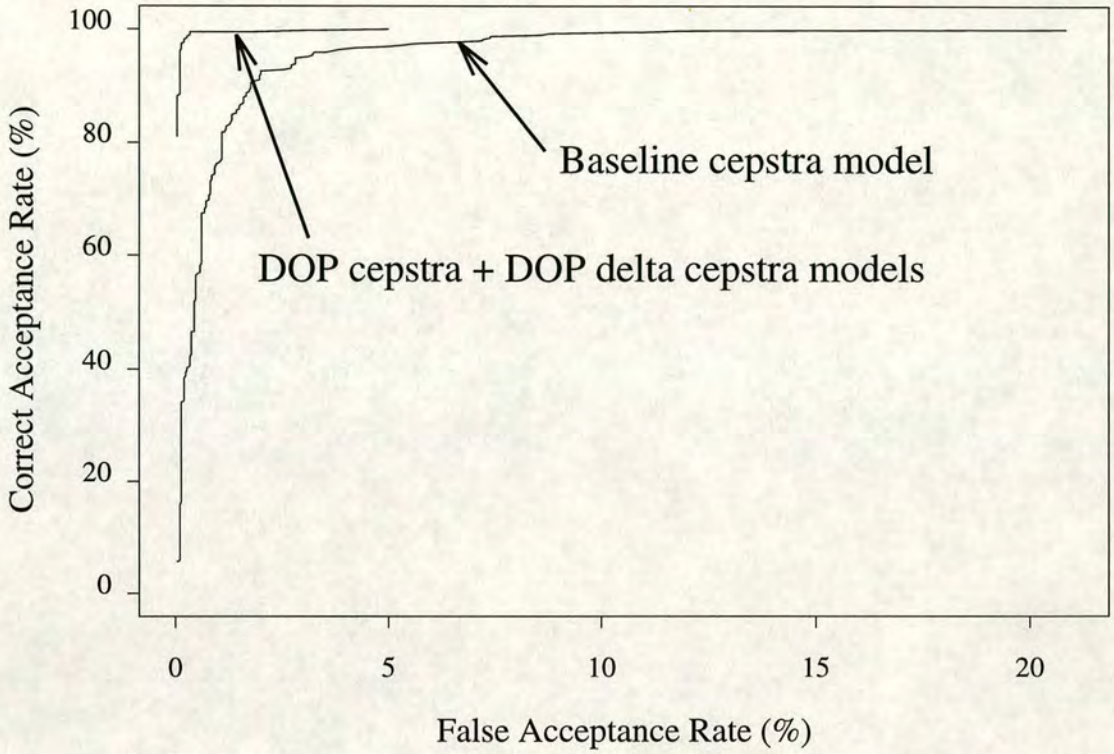

Figure 5.8: Receiver operating characteristic for the DOP cepstra plus Δcepstra model combination and the baseline cepstra models. A SS EER is used on the 12-digit-sequence.

Figure 5.9 shows the impact of DOP modelling relative to the algorithms of Chapter 4. There is a clear line of development beginning with the baseline cepstra, then adding a second feature set, then using DOP models and finally using DOP models with two feature sets. The top figure plots the SS EER for the different digit sequences while the lower figure plots the equivalent TDM values. The TDM values are interesting because they are not bounded in the same way that the EER is and so the relative performance of the different systems can be seen in an alternative, and perhaps more meaningful way.

DOP models with their increased discriminating power clearly out-perform the conventional models used in Chapter 4. There is no advantage in using the conventional models, even in combination with the DOP models. Substantial benefits of the order of 54-75% can be gained by using DOP models with two feature sets. The best feature set combinations are those involving a static and a delta feature set. A slight advantage can be gained from using cepstra, Δcepstra , MFCC and ΔMFCC information streams with equal weights.

It is interesting to note that the use of DOP models does not eliminate the benefits of using speaker specific thresholds. The use of discriminative scoring helps to stabilise thresholds but

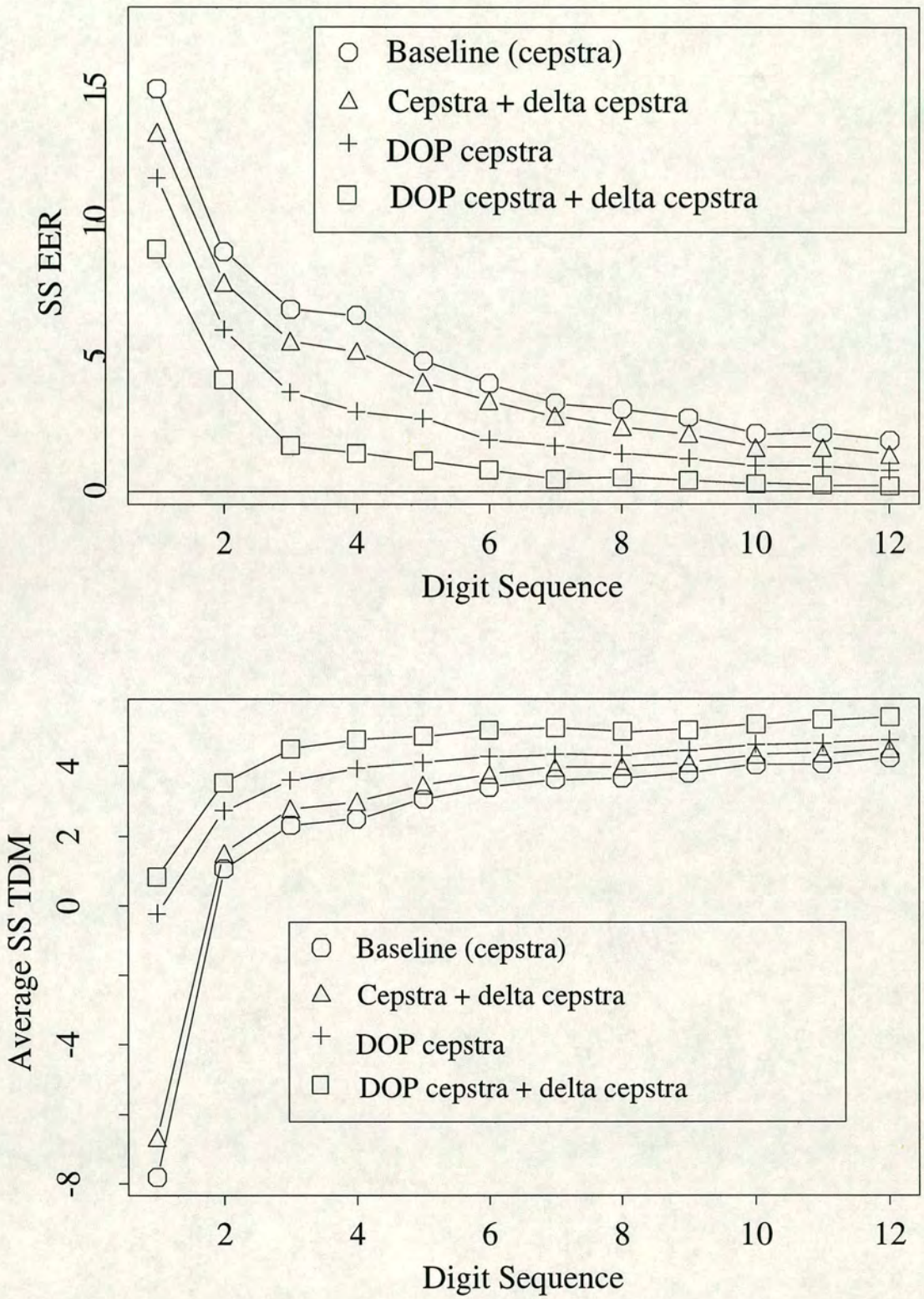

Figure 5.9: Summary of the main progressions in algorithms using SS TDM for the different sequence lengths.

the SS EER is still 81% less than the SI EER for the DOP system using cepstra and Δcepstra feature sets. DOP models are superior to speaker normalisation, and this is due to the use of a consistent state segmentation. The DOP model architecture offers several opportunities for further development, in particular the development of a better discriminating function and the use of alternative feature sets.

In summary the use of DOP HMMs for ASV has the following benefits.

- The use of DOP increases the speaker discriminating power of HMMs for ASV.
- DOP modelling can be easily added to a conventional HMM ASV system, and DOP models can be derived from conventional HMMs with no extra training.
- The extra calculation needed to create a DOP model is very small and occurs after feature extraction and state segmentation, so the effect on the overall calculation time is minimal.
- Model adaptation is not affected by the use of DOP models. Both client and impostor models can be adapted in the same ways as a conventional HMM.
- The impostor model can be easily changed and improved without the need for retraining all the speaker specific models, as would be the case with discriminative training approaches.
- The explicit separation of speech and speaker modelling gives DOP models a theoretical advantage over discriminative training approaches, although this has not been experimentally verified.
- DOP out-performs the speaker-normalisation technique because the use of a single state segmentation facilitates more meaningful frame by frame comparisons of observation probabilities. DOP modelling also requires significantly less computation than speaker normalisation because only one state segmentation is performed instead of two[8].

[8] Assuming a single impostor model. If cohort speakers are used the benefit is even greater.

Chapter 6

Summary and Conclusions

The motivation for this research comes from applications such as telephone banking which can be greatly enhanced by reliable automatic speaker verification technology. This research was started in 1991, when HMMs had begun to emerge as the best available architecture for text-dependent ASV applications (Rosenberg *et al.*, 1990b). The best way to use HMMs for ASV was, and still is, an open question.

Much work had already been done investigating the use of HMMs for ASR applications, and these speech recognition systems could be successfully applied to the ASV task (Rosenberg *et al.*, 1990b; Rosenberg *et al.*, 1990a; Rosenberg *et al.*, 1991). Although closely related, ASV and ASR are different tasks. This program of research has focused on the differences between the two tasks. The key advantage of text-dependent ASV over ASR is that, because the text is known, ASV systems are far less computationally expensive. This was exploited in HASAS by using the more accurate, but more computationally demanding Gaussian state duration modelling. The key disadvantage of ASV compared to ASR is that models must be speaker dependent by definition and so the amount of training data is strictly limited, and is far less than is available for speaker independent ASR. This restriction led to the use of SCHMM. The Gaussian mixtures offer better modelling than DHMM, while the use of a codebook reduces the number of parameters to be estimated during training compared to CHMMs.

Another, practical, disadvantage is that standard databases and tasks for ASV did not exist until very recently (Campbell, 1995). ASR research has benefited greatly from the availability of standard tasks and databases. The lack of a common task has made it difficult to evaluate the importance of new algorithms and slowed the progress of the field.

The main difference in objective between ASV and ASR is that ASV requires maximum

inter-speaker variation and minimum intra-speaker variation for a given utterance. ASR seeks to minimise both inter-speaker and intra-speaker variation. This difference strongly suggests that the feature extraction process should be different for ASV and ASR. Feature extraction is a coding process. The total amount of data is reduced while retaining all information useful to the task. Many different types of information are coded into the speech signal. Apart from lexical information there is information about the emotional state of the speaker, whether they are stressed, excited, tired and so on. There is also information about the speaker -the physical size and shape of their vocal apparatus, their sex and their accent.

Since the ideal feature set extracts all useful information but discards all other sources of variability, the same feature set cannot possibly be ideal for both ASV and ASR. The difficulty in selecting a good feature set for ASV is that speaker dependent models are both speech and speaker models, and so the feature set must be a compromise which is suitable for both tasks. It was with this fact in mind that the HASAS architecture was designed. The speech modelling (the state segmentation) was separated from the speaker modelling (the calculation of the verification score) during both training and testing. This allows the speech modelling and the speaker modelling stages of the verification process to use different feature sets. Selecting the best feature set for the speech modelling is an ASR problem. The cepstral feature set was used as a standard throughout this work, but could, and should, be replaced by whatever feature sets prove to be better for ASR tasks.

The work in Chapter 4 focused on investigating the relative merits of four commonly used feature sets for the speaker modelling stage of the verification process. The four feature sets were LPC cepstra, Δcepstra, MFCC and ΔMFCC.

Given the state segmentation[1] the average observation probability per frame is calculated and used as the verification score. Note that the duration probabilities are not included in the verification score in order to fully isolate the speaker discriminating power of each feature set. Considerable variation in the performance of the feature sets was observed, with the best SS EER of 1.93%[2] being produced using the LPC cepstra feature set.

The MFCC features did not perform as well as the LPC cepstra. Another important point is that the Δfeatures did not perform as well as the static features. This is surprising in view of the

[1] From a Viterbi search using the cepstra feature set.

[2] All results quoted in this chapter are for a 12-digit sequence test utterance.

fact that Δ features are naturally more robust to channel variation because they do not rely on absolute values.

Other researchers have successfully improved ASV performance by combining cepstra and Δcepstra but it is not clear in text-dependent applications how much the improvement is due to improving the state segmentation and how much is due to improving speaker discrimination. The HASAS architecture allows this to be tested. The four feature sets each produce a verification score based on the same state segmentation. Two feature sets can be used by combining their scores in a weighted sum. By adjusting the weight an optimum combination of scores can be experimentally determined. This was done for each of the six possible pairwise combinations of feature sets.

This was found to improve SS EERs from 4-36% compared to the use of a single feature set. The best SS EERs of 1.36% and 1.39% were obtained by combining cepstra with Δcepstra and with MFCC respectively. The combination of the two Δ features sets was not very successful, with only 4% improvement and an SS EER of 4.25%. The combination of three features sets (cepstra, Δcepstra and MFCC) using equal weights produced a further reduction in SS EER to 1.22%.

This result supports an approach of using as many different information sources as possible, within the bounds of computational feasibility. The combination of information streams need not be done using a simple linear weighted sum. A more optimal combination of multiple information streams is a task for which a neural network would seem well suited. A neural network could capture any subtle inter-dependencies in the scores of partially correlated feature sets.

The usefulness of the state duration probabilities as an information stream was investigated in Chapter 4. These probabilities were shown to contain speaker discriminating information - performing better than the observation probabilities from the ΔMFCC feature set. However, when combined in a linear sum with the scores from the best pair of feature sets (cepstra and Δcepstra), the state duration information produced only minor improvement in the SS and SI EERs[3].

The combination of more diverse, and potentially less correlated information streams is recommended as a research area with great potential. Features such as energy, formant frequencies,

[3] 1% and 6% respectively

and second order difference cepstra would be interesting candidates for investigation.

The combination of multiple information streams is motivated by the desire to make maximum use of the information produced by the HMMs. Consistent with that approach, experiments were conducted into digit weighting in Chapter 4. There is much experimental evidence that different phones have different amounts of speaker discriminating information[4]. The variation in the ASV performance of the 12 digits when single digit test utterances were used suggested that it might be useful to weight the digit scores when combining them to get 12-digit sequence scores. The digit weights were proportional to their relative performance in single digit tests on the assumption that this would be representative of their speaker discriminating power. This assumption ignores the effect of correlations in the information from the different digits. This approach to digit weighting produced only a minor improvement in EER.

It is intuitively reasonable, and supported by practical experience, that *different* speakers can best be identified on the basis of *different* speech sounds. When recognising people, for example, we tend to focus on certain unusual features which distinguish them. For one person we might look at their hair, for another their height. It is reasonable to consider the possibility that we recognise a person's speech in the same way,and that the relative importance of different words or phones is speaker specific. Further investigation showed this to be the case. By making the digit weights speaker specific, the SS EER for the LPC cepstra feature set was reduced from 1.93% to 1.29%. This indicates that another useful avenue for further research might be the adaptive training of speaker-specific word or phone weights. It may also be possible to construct optimal password phrases which are individually designed to maximise discrimination for a given client.

6.1 DOP Modelling for ASV

ASV is a binary classification task. Utterances must be classified as either originating from the client speaker or from an impostor. In Chapter 4 only the client class was modelled. The absolute value of the likelihood *match* of the utterance with the client model is used as a verification score. If the match is good the utterance is classified as belonging to the client. If the match is bad, for whatever reason, the utterance is classified as belonging to an impostor. The client model is a

[4] Refer to Section 4.3 for references

model both of the speaker and of the utterance, however, and so variation in the likelihood score of the model can be due to factors other than the speaker's identity. This makes classification based on the client model score very sensitive to such things as channel or handset variation. The use of the client model likelihood as a verification score is termed the client model (CM) approach. In hindsight it seems obvious that a better approach is to use two class models -a model of the client class and a model of the impostor class. This is termed a client-impostor model (CIM) approach. Using the CIM approach the verification decision is based on whether the utterance is better classified as a client utterance or an impostor utterance. This means that variation in the quality of the speech (as manifested in a variation in the speech recognition score of the utterance) do not affect the classification decision. The CIM approach is referred to in the literature as speaker/cohort/similarity normalisation. The CIM approach has proven to be very successful, but the classification decision is still external to the modelling process.

Ideally a discriminative model should be used which models the difference between the client speaker and all other speakers. Some methods of discriminative training for HMMs have been proposed[5] and these have been successful for ASR. The difficulty with using discriminatively trained HMMs for ASV is the fact that text-dependent ASV is both a speech and a speaker recognition task. If the client model is trained to model the differences in the client's speech compared to other speakers, it will no longer be an optimal *speech* model.

The separation of the speech and speaker modelling processes in the HASAS architecture, on the other hand, allows the speaker model to be speaker discriminating while the speech model is still trained for speech recognition. The discriminating model is constructed by contrasting client and impostor models. Some discriminating function $\mathcal{F}_{\mathrm{DOP}}$ is used to contrast the observation probability surfaces of the two models to produce a new *discriminating* observation probability surface (DOP) which is used in the verification calculation (the *speaker* modelling stage of the verification process).

For DOP modelling to be successful the speech and speaker models must be compatible. Specifically, the equivalent states in the two models must correspond to the same acoustic events. This is encouraged by seeding the training of the client models with the speaker independent impostor models. Chapter 5 is an investigation of DOP modelling and its application to ASV in particular.

[5] Refer to Section 2.4.3 for references

DOP modelling was evaluated using each of the feature sets used in Chapter 4, and was extremely successful. Improvements in EER of 43-90% were obtained over the CM approach of Chapter 4. The combination of the two feature sets proved to be even more rewarding than it had been using the CM approach, with improvements in SS EER of 38-78% compared to the use of a single feature set. The best pair-wise combination of feature sets produced an 85% reduction in SS EER compared to the best pair-wise combination of features for the CM approach.

Using DOP models, all six pairwise combinations of feature sets produced comparable performance, with SI EERs ranging from 1.1% to 1.64%. This indicates the DOP modelling is robust to the choice feature set. In particular, the combination of Δcepstra and ΔMFCC, which produced a poor SS EER of 4.25% using the CM approach showed a reduction of 94% to 0.26% when DOP models were used.

The DOP models were compared to the CIM (speaker normalisation) approach in Section 5.8 and proved to be superior in all cases. The SS EER using the best pair of feature sets was over 50% less for DOP than for the CIM approach. The difference in errors between DOP and CIM using the cepstra and Δcepstra feature sets had a significance level of 0.013. It was concluded from this that DOP models should be used in place of the CIM approach.

Although DOP models have proven to be very successful on the ASV task, the architecture is extremely flexible and there are several areas where further development might be rewarded. In particular the choice of impostor model, the choice of segmentation model, and the choice of discriminating function $\mathcal{F}_{\mathrm{DOP}}$. There is also the application of DOP modelling to other classification tasks to be considered.

The choice of impostor model has already been studied to some extent with reference to the CIM approach and was discussed in Section 2.4.4. It is not yet clear from studies in the literature what the best form of impostor model is but a well-trained speaker-independent model, as used in Chapter 5 appears to be a good choice.

The choice of segmentation model was investigated in Section 5.6. Two possibilities were considered, namely the client model and the impostor model. Both models have potential advantages - the speaker independent impostor model is trained with more data but the client model is *tuned* to the client's speech and the use of the client model for segmentation is more consistent with the implicit state segmentation in the training. The client model proved experimentally to be the superior segmentation model in almost all cases. It is possible, however,

that a better approach would be to train a separate set of models specifically for segmentation. These could be constructed by adapting the speaker independent model to the client to create speaker dependent segmentation models. This approach may combine the robustness of the speaker independent model with the speaker-specific benefits of a speaker dependent model.

The discriminating function $\mathcal{F}_{\mathrm{DOP}}$ which is used to construct the DOP model from the client and impostor models is a simple likelihood ratio. The use of this function is based on the assumption of robustly trained client and impostor models. The discussion in Section 5.9 pointed out reasons why a likelihood ratio is not a good choice if this assumption is not correct. An alternative function could be constructed which is based on experimental analysis of the observation probabilities generated by the client and impostor models. A neural network could be trained to produce such a function, taking as inputs the client and impostor observation probabilities for a given frame, perhaps from several different feature sets, and producing a single (DOP) output.

A focus on the differences between ASR and ASV has been the key element of this research. It has led to the use of SCHMMs which can perform well with limited amounts of training data, and to the use of state duration modelling. It is also behind the separation of speech and speaker modelling in the HASAS architecture. The separating of speech and speaker modelling has, in turn, allowed a new form of discriminative modelling known as DOP modelling to be used. DOP models have been successfully tested on the ASV task and they have been shown to be superior to the speaker normalisation technique currently favoured in the literature. The DOP architecture is a powerful and flexible architecture which can potentially be applied to other binary classification problems.

Appendix A

Statistical Significance Tests

A.1 Comparing Two Algorithms

Chapters 4 and 5 present many different algorithms, all of which are evaluated on the same database. Although it is generally clear which algorithm is better, a measure of the significance of the result is required.

In the following test speaker specific thresholds are used. The reasoning for this is that errors caused by sub-optimal thresholds are not *directly* related to the algorithm and can, perhaps, be corrected by making the threshold speaker specific. The following approach is taken to performing a statistical significance test to the results from two algorithms (A and B). It is based on McNemar's test (McNemar, 1947), as described in (Gillick & Cox, 1989).

1. Only the *a block* results are used, i.e. training on *a* block and testing on *b, c, d and e* blocks. This ensures independence of the trial data.

2. The 12-digit-sequence scores are used.

3. For each client an EER threshold is calculated and subtracted from that clients scores.

4. The FR rate is set to a fixed value by specifying the number of FR errors over all clients, thus determining a threshold for each of the algorithms (τ_A and τ_B). The FR rate is chosen to be as close as possible to the lower of the SS EERs for the two algorithms.

5. The thresholds are applied to obtain a decision for each trial, for each of the algorithms. For each trial there are 4 possibilities.

A	correct	B	correct	
A	incorrect	B	incorrect	(common errors)
A	incorrect	B	correct	(A!B errors)
A	correct	B	incorrect	(!AB errors)

The total number of FR errors will be the same for A and B, as defined by step 4. The number of common errors is also the same by definition. The number of A!B and !AB errors can differ and it is this difference that is the subject of the significance test.

We define the following.

n_{10}=	number of A!B	FA errors
n_{01}=	number of !AB	FA errors
H_0=	*Algorithms A and B have equivalent performance.*	

If the null hypothesis (H_0) is true then n_{10} and n_{01} will be binomially distributed relative to $k = n_{10} + n_{01}$. We wish to determine the probability of sampling values *at least as different* as n_{10} and n_{01} if H_0 is true.

The usual definitions for binomial distributions are as follows.

m=	number of successes
k=	number of trials
$q = \frac{1}{2}$	probability of success
R	Random variable

$$P(R = m|k, q) = \frac{k!}{m!(k-m)!}\left(q^m(1-q)^{(k-m)}\right) \tag{A.1}$$

For this test $k = n_{10} + n_{01}$ and $q = 0.5$

Using a two-tailed test and choosing the labels A and B such that $n_{10} <= n_{01}$ the significance level α is given by Equation A.3.

$$\begin{aligned} \alpha &= \sum_{m=0}^{n_{10}} P(R = m|k, q) + \sum_{m=n_{01}}^{k} P(R = m|k, q) && n_{10} < n_{01} \\ \alpha &= 1 && n_{10} = n_{01} \end{aligned} \tag{A.2}$$

This significance test is only quoted where there might be doubt about the significance of a result. Experiments which show clear improvement do not have significance levels quoted. Note that because n_{10} and n_{01} are small (generally around 10), significance levels cannot be

very high. This is unavoidable, and is a result of trying to differentiate algorithms on the basis of errors when neither algorithm produces many errors. The targeted distance measure, although it is only an ad-hoc measure, is probably at least as useful for low error rates because it takes into account near-misses as well as errors and thereby gives a measure of the robustness of the algorithm. Unfortunately a significance test for the TDM has not yet been formulated.

Appendix B

Summary Tables for Chapter 4

B.1 Description of Summary Tables

Detailed result summaries are presented here for the various algorithms discussed in Chapter 4. the result summaries for the algorithms discussed in Chapter 5 are in a separate appendix.

Each result summary consists of 3 tables. The first is a summary of the results for each of the 12 digits. There are 4 performance measures quoted. The first two, labelled *SI EER* and *SS EER* are the equal error rate (EER) using speaker independent (SI) and speaker specific (SS) thresholds. The results for the speaker specific thresholds are averaged over all speakers. The other two performance measures are targeted distance measures (TDM), again using speaker independent (SI) and speaker specific (SS) thresholds. Refer to Section 2.2 for more detail on the performance measures used.

The second table gives results for the digit sequences. This table uses the same 4 performance measures as the first table but includes two additional performance measures - the zero false rejection rate (ZFR) and zero false acceptance rate (ZFA). The ZFR and the ZFA are both calculated using speaker specific thresholds, and are averaged over all speakers. The use of the digit sequences is described in Section 3.4.11. Note that a digit sequence of length 1, is the same as using the digit *one* in isolation, so the first row of this table will contain the same values given in the first row of the previous table.

The third table looks in more detail at the results for the 12-digit-sequence, with a break-down of errors by client. All performance measures are using speaker specific thresholds. Note that the mean values quoted in the final row of the table are the same values given for the 12-digit-sequence in the previous table.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	16.91	15.2	-3.61	-7.81
two	14.9	12.99	-0.98	-0.57
three	16.59	14.18	-1.33	-0.88
four	24.6	21.57	-6.29	-6.53
five	13.12	12.14	-0.44	-1.38
six	15.88	14.07	-1.25	-0.97
seven	11.82	9.89	0.51	1.07
eight	15.36	12.88	-1.29	-0.82
nine	11.73	10.13	0.46	1.01
zero	13.08	10.86	0.23	0.81
nought	17.3	15.06	-2.3	-2.24
oh	14.32	11.94	-0.71	-0.28
mean	15.47	13.41	-1.42	-1.55

Table B.1: LPC Cepstra. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	16.91	15.2	-3.61	-7.81	62.41	88.9
2	11.15	9.08	0.61	1.05	45.19	66.48
3	8.43	6.91	1.87	2.3	36.44	57.9
4	8.97	6.68	1.88	2.48	29.27	54.05
5	7.17	4.95	2.48	3.06	22.04	47.71
6	6.21	4.09	2.82	3.39	14.91	40.67
7	5.43	3.34	3	3.62	12.15	33.1
8	5.4	3.13	3.03	3.66	11.04	35
9	5.11	2.8	3.19	3.8	10.37	31.38
10	4.1	2.19	3.46	4.04	8.79	24.71
11	4.21	2.22	3.45	4.06	8.82	22.33
12	3.69	1.93	3.6	4.23	7.72	18

Table B.2: LPC Cepstra. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.66	0	0
2	0.4	4.38	0.8	4
3	2.1	4.63	2.2	49
4	0.8	4.24	3.2	8
5	1.3	4.79	2.4	6
6	0.4	5.1	0.8	4
7	0.4	4.16	0.8	10
8	2.7	3.92	5.2	23
9	0.2	4.85	0.4	1
10	0.5	4.33	1	3
11	4.3	2.85	37.2	12
12	0	4.65	0	0
13	3.3	3.86	8.8	22
14	0.4	4.93	0.8	15
15	1.2	5.06	1.8	4
16	5.4	2.71	12.4	56
17	2.1	4.68	2.8	50
18	0.8	5.34	1.6	5
19	6.7	2.25	37	66
20	6.4	2.11	41.6	28
21	1.2	4.42	1.4	12
mean	1.93	4.23	7.72	18

Table B.3: LPC Cepstra. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	24.25	22.27	-8.07	-16.19
two	27.75	24.43	-9.44	-11.85
three	22.12	19.5	-4.46	-4.09
four	36.05	34.12	-22.61	-34.99
five	23.95	22.65	-5.86	-18.83
six	31.07	27.81	-12.66	-13.01
seven	25.81	23.46	-6.3	-5.69
eight	20.91	19.08	-4.17	-5.5
nine	20.89	16.98	-3.23	-2.99
zero	23.17	19.22	-5.05	-3.53
nought	24.25	21.86	-6.5	-8.87
oh	32.52	29.2	-16.2	-27.93
mean	26.06	23.38	-8.71	-12.79

Table B.4: LPC ΔCepstra. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	24.25	22.27	-8.07	-16.19	74.98	93.76
2	20.33	17.1	-4.05	-4.15	59.23	90.9
3	15.62	12.02	-1.06	-0.15	42.58	78.1
4	15.58	12.06	-1.03	-0.39	40.97	74.19
5	13.83	9.96	-0.04	0.93	34.27	66
6	13.61	8.87	0.21	1.41	29.93	58.19
7	12.11	7.48	0.68	1.95	24.55	51.48
8	10.55	6.6	1.17	2.33	19.53	43.86
9	10.37	6	1.36	2.59	16.13	44
10	10.11	5.01	1.54	3	13.66	35.29
11	9.36	4.56	1.74	3.26	13.24	30.38
12	9.75	4.4	1.59	3.3	13.3	29.33

Table B.5: LPC ΔCepstra. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0.9	4.82	3.2	8
2	0	4.76	0	0
3	4.6	3.02	20.6	56
4	0	4.76	0	0
5	6.1	3.26	19.6	27
6	0	5.06	0	0
7	0.4	4.78	0.8	1
8	0.4	4.9	0.8	1
9	1.3	4.42	5	8
10	0	5.58	0	0
11	15.1	-0.12	45.8	65
12	0	4.71	0	0
13	4.7	2.43	21.6	57
14	3.9	3.32	16.4	17
15	2.4	3.26	8.8	84
16	11.3	1.63	18.6	56
17	2.6	3.59	8.6	28
18	0.1	5.7	0.2	2
19	12.3	0.87	29.6	59
20	21.3	-3.94	66.2	100
21	5.1	2.43	13.4	47
mean	4.4	3.3	13.3	29.33

Table B.6: LPC ΔCepstra. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	21.63	19.7	-7.37	-22.74
two	14.81	12.98	-1.19	-1.02
three	16.72	14.59	-1.45	-1.12
four	24.59	19.94	-5.48	-4.5
five	14.92	12.65	-1.1	-0.67
six	17.44	14.68	-2.19	-2.36
seven	14.79	11.78	-0.78	0.11
eight	16.87	15.08	-1.99	-1.98
nine	14.41	11.94	-0.89	-0.25
zero	14.4	13.1	-1.05	-0.68
nought	15.83	12.88	-1.5	-1.09
oh	15.04	13.24	-1.28	-4.23
mean	16.79	14.38	-2.19	-3.38

Table B.7: MFCC. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	21.63	19.7	-7.37	-22.74	63.82	94.81
2	11.5	10.11	-0.05	-0.41	44.7	69.71
3	9.63	7.52	1.29	1.79	36.23	59.52
4	9.54	6.76	1.55	2.3	31.53	59.95
5	8.78	5.38	1.82	2.82	24.66	43.95
6	8.2	4.77	2.14	3.11	22.14	37.1
7	7.54	4.27	2.28	3.36	17.84	30.95
8	7.08	4.31	2.33	3.38	15.46	31.05
9	6.71	3.76	2.51	3.56	14.96	28.57
10	5.97	3.21	2.79	3.81	12.98	27.71
11	5.64	3.01	2.92	3.94	12.46	23.38
12	5.08	2.59	3.11	4.13	11.51	21.05

Table B.8: MFCC. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0.1	5.38	0.2	1
2	1.2	4.41	1.6	20
3	2.5	4.31	4.2	35
4	8.4	1.5	43	53
5	0.3	5.37	0.6	7
6	0.1	5.48	0.2	1
7	3.6	3.41	6.4	22
8	5.4	2.7	11.8	62
9	1	4.39	4.4	4
10	0	4.94	0	0
11	8.2	2.16	30	23
12	0	5.39	0	0
13	0.8	5.53	1.6	2
14	1.6	4.78	3.8	5
15	0.4	5.11	0.8	6
16	1.3	4.72	2.6	21
17	0.1	4.98	0.2	17
18	2.6	4.07	5.8	28
19	7.8	1.37	60.8	69
20	9	1.54	63.8	66
21	0	5.13	0	0
mean	2.59	4.13	11.51	21.05

Table B.9: MFCC. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	31.21	28.34	-15.53	-40.27
two	26.49	23.4	-10.17	-15.4
three	29.16	24.7	-10.04	-13.68
four	35.53	35.16	-22.9	-28.28
five	29.73	27.86	-10.81	-11.88
six	33.64	30.56	-21.7	-38.69
seven	28.8	26.06	-10.57	-14.95
eight	32.63	27.13	-15.2	-235.75
nine	27.48	24.14	-9.24	-22.67
zero	28.17	25.23	-10.62	-13.33
nought	26.96	23.56	-9.51	-12.55
oh	30.26	25.47	-11.86	-21.49
mean	30	26.8	-13.18	-39.08

Table B.10: ΔMFCC. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	31.21	28.34	-15.53	-40.27	81.52	97.67
2	23.92	19.8	-7.13	-8.44	68.59	95.43
3	22.14	16.62	-5.17	-5.39	53.9	93.29
4	21.59	16.75	-4.72	-6.05	54.49	93.57
5	20.09	15.57	-3.88	-3.04	50.42	91.57
6	20.91	14.79	-4.18	-5.19	46.9	86.71
7	19.95	13.5	-3.57	-3.09	41.34	85.29
8	20.54	13.23	-3.96	-20.96	36.96	85.05
9	20.3	12.21	-3.59	-5.85	34.15	78.95
10	19.31	11.07	-3.1	-1.56	32.84	78
11	18.2	9.93	-2.4	0.11	30.18	72.52
12	17.84	9.53	-2.23	0.28	28.71	71

Table B.11: ΔMFCC. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	3.3	4.08	5.6	98
2	3.4	3.83	7.2	96
3	4.5	3.33	14.4	36
4	3.1	3.34	4	18
5	9.7	1.47	26.6	69
6	4.7	2.67	12	75
7	9.2	1.87	30.4	87
8	4.5	3.17	8.4	35
9	2.8	4.29	4.6	68
10	1.1	5.14	2.2	20
11	12.4	0.55	53.8	81
12	1.5	4.42	2.2	6
13	24.2	-5.84	71.6	100
14	10.5	0.63	45.6	94
15	9.1	1.52	26.6	70
16	20.3	-3.59	80.2	100
17	11.3	1.01	31	88
18	6.2	3.07	13.8	57
19	12.2	0.56	33.6	95
20	38.7	-31.4	97	99
21	7.4	1.7	32.2	99
mean	9.53	0.28	28.71	71

Table B.12: ΔMFCC. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	31.91	28.4	-12.14	-16.96
two	26.81	22.38	-7.79	-7.63
three	26.41	22.71	-7.67	-12.92
four	28.77	26.63	-9.16	-21.15
five	22.22	17.79	-4.27	-23.89
six	32.18	26.4	-12.4	-14.15
seven	19.82	15.46	-2.86	-2.16
eight	25.98	20.66	-8.01	-18.56
nine	16.21	13.61	-1.35	-0.73
zero	22.39	19.22	-5.76	-14.37
nought	26.46	20.39	-7.66	-10.7
oh	20.05	17.77	-3.43	-9.87
mean	24.93	20.95	-6.88	-12.76

Table B.13: State Duration Probability Φ_{DUR}. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	31.91	28.4	-12.14	-16.96	75.07	98.33
2	25.21	20.3	-6.16	-4.98	61.21	94.29
3	22.97	18	-4.4	-3.65	50.74	90.43
4	21.39	17.06	-3.57	-3.07	44.85	92.05
5	18.99	13.79	-2.26	-1.82	38.06	90.67
6	19.68	13.81	-2.51	-4.18	35.92	91.14
7	17.85	12.12	-1.57	-1.29	30.01	88
8	17.94	11.61	-1.57	-1.25	28.42	86.57
9	15.63	10.26	-0.67	0.72	25.78	85.81
10	14.87	9.38	-0.42	1.37	24.5	84.67
11	15.07	9.07	-0.48	1.53	23.66	79.71
12	14.57	8.44	-0.17	1.84	20.95	79.05

Table B.14: State Duration Probability Φ_{DUR}. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0.8	4.88	1.6	28
2	3.1	3.91	4.2	84
3	1.9	4.8	2.8	65
4	2.5	3.42	9.8	100
5	13.8	-0.36	32.6	100
6	4.4	3.22	17.6	94
7	18.4	-2.2	39	100
8	11.6	0.58	28.4	100
9	2.4	4.14	3.2	88
10	12.8	0.21	30.2	100
11	1.3	4.43	2	22
12	2.1	4.42	2.2	97
13	22.2	-4.61	68.4	96
14	7.5	2.38	17	96
15	1	4.99	2.6	2
16	10.5	1.54	17.2	96
17	17.9	-1.55	58.8	100
18	2.1	4.3	4.4	30
19	9.4	1.31	23.8	100
20	15.5	-0.34	45.2	62
21	16	-0.73	29	100
mean	8.44	1.84	20.95	79.05

Table B.15: State Duration Probability Φ_{DUR}. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	15.55	13.55	-2.83	-6.67
two	13.58	11.83	-0.5	-0.1
three	13.63	12.08	-0.25	0.19
four	23.72	21.08	-6.03	-6.59
five	12.42	11.37	-0.04	-0.52
six	14.76	12.67	-0.75	-0.39
seven	10.47	8.59	1.04	1.47
eight	14.79	11.24	-0.8	-0.19
nine	11.12	8.92	0.81	1.42
zero	10.86	8.98	1.01	1.59
nought	15.5	13.38	-1.25	-1.17
oh	13.62	11.33	-0.49	-0.13
mean	14.17	12.09	-0.84	-0.92

Table B.16: Cepstra plus Δ Cepstra (α = 0.6). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	15.55	13.55	-2.83	-6.67	57.72	84.67
2	10.06	7.93	1.04	1.5	40.82	59.9
3	7.39	5.7	2.34	2.78	30.08	52.38
4	7.58	5.32	2.34	2.98	23.42	47.38
5	6.23	4.14	2.83	3.45	18.1	40.95
6	5.27	3.42	3.18	3.77	11.79	32.05
7	4.62	2.84	3.38	3.98	9.95	25.81
8	4.5	2.44	3.36	3.99	8.94	26.57
9	4.11	2.19	3.47	4.1	8.81	23.86
10	3.34	1.68	3.73	4.34	7.36	17.24
11	3.33	1.67	3.75	4.38	7.18	15.33
12	3.08	1.4	3.84	4.51	6.41	13.43

Table B.17: Cepstra plus Δ Cepstra (α = 0.6). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.82	0	0
2	0.2	4.61	0.4	1
3	1.4	4.77	2.2	37
4	0.1	4.69	0.2	1
5	1.3	5.04	1.8	6
6	0	5.49	0	0
7	0	4.61	0	0
8	0.6	4.46	1.2	13
9	0.2	5.11	0.4	1
10	0.1	4.77	0.2	1
11	4.2	2.66	37.2	15
12	0	4.97	0	0
13	2.2	4.56	4.8	13
14	0.3	4.95	0.6	6
15	1.2	5.11	1.6	4
16	3.4	2.97	10.2	41
17	2.1	5.08	2.2	41
18	0.1	5.88	0.2	3
19	5.2	2.55	32.4	64
20	6.5	2.02	38.4	25
21	0.3	4.61	0.6	10
mean	1.4	4.51	6.41	13.43

Table B.18: Cepstra plus Δ Cepstra ($\alpha = 0.6$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	15.39	13.84	-3.15	-8.73
two	12.02	10.63	0.22	0.43
three	14.15	12.27	-0.45	-0.01
four	22.3	18.45	-4.49	-3.94
five	11.83	10.57	0.23	-0
six	13.9	11.87	-0.53	-0.22
seven	10.16	8.09	1.04	1.7
eight	13.89	11.6	-0.71	-0.18
nine	10.36	8.36	0.96	1.63
zero	10.84	9.19	0.88	1.45
nought	13.55	11.69	-0.62	-0.46
oh	11.92	9.78	0.12	0.22
mean	13.36	11.36	-0.54	-0.67

Table B.19: Cepstra plus MFCC (α = 0.7). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	15.39	13.84	-3.15	-8.73	56.37	87.57
2	8.7	7.21	1.42	1.57	40.52	50.38
3	6.89	5.6	2.38	2.73	32.93	47.52
4	6.92	5.47	2.51	2.99	25.89	43
5	5.96	4.08	2.88	3.47	19.48	35.43
6	5.22	3.42	3.15	3.74	14.09	30.62
7	4.48	2.58	3.27	3.93	11.14	23.76
8	4.5	2.58	3.28	3.93	9.79	23.67
9	4.13	2.24	3.43	4.07	9.51	21.52
10	3.53	1.8	3.66	4.29	7.94	17.86
11	3.37	1.74	3.72	4.35	7.56	15.43
12	2.93	1.41	3.85	4.5	6.68	13.33

Table B.20: Cepstra plus MFCC (α = 0.7). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.8	0	0
2	0.5	4.54	1	6
3	1.3	4.75	1.6	45
4	1.6	3.89	3.2	17
5	0.3	5.25	0.6	2
6	0	5.42	0	0
7	0.3	4.32	0.6	8
8	1.4	3.89	2.2	41
9	0.5	4.92	1	1
10	0.1	4.6	0.2	2
11	5.6	2.84	31.8	12
12	0	5.06	0	0
13	0.9	4.78	2.6	5
14	0.4	5.21	0.8	8
15	0.1	5.32	0.2	1
16	1.8	3.76	4.6	10
17	1.3	5.43	1.8	22
18	1.1	5.4	1.2	6
19	5.6	2.27	45.8	68
20	6.9	2.35	41	26
21	0	4.81	0	0
mean	1.41	4.5	6.68	13.33

Table B.21: Cepstra plus MFCC ($\alpha = 0.7$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	16.18	14.65	-3.36	-7.64
two	14.06	11.83	-0.66	-0.28
three	14.98	13.39	-0.92	-0.47
four	23.99	21.07	-5.88	-6.24
five	12.41	11.61	-0.2	-0.79
six	15.7	13.67	-1.14	-0.79
seven	11.66	9.35	0.64	1.25
eight	14.42	12.08	-1.02	-0.54
nine	11.46	9.55	0.7	1.19
zero	12.44	10.3	0.43	1.06
nought	16.23	13.98	-1.77	-1.58
oh	13.85	11.11	-0.46	-0.06
mean	14.78	12.72	-1.14	-1.24

Table B.22: Cepstra plus Δ MFCC ($\alpha = 0.7$). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	16.18	14.65	-3.36	-7.64	60.45	87.05
2	10.52	8.46	0.84	1.28	42.76	60.67
3	8.25	6.48	1.97	2.48	34.07	57.43
4	8.5	6.16	2.05	2.67	27.91	50.67
5	6.77	4.63	2.6	3.2	20.66	47.81
6	6.1	3.96	2.9	3.51	14.06	39.48
7	5.25	3.11	3.07	3.74	11.58	30.71
8	5.12	2.82	3.09	3.78	10.26	32.43
9	4.7	2.55	3.27	3.92	9.79	28.86
10	3.84	2.07	3.52	4.15	8.04	22.9
11	3.92	2	3.56	4.19	7.98	20.86
12	3.44	1.69	3.68	4.36	7.16	16.24

Table B.23: Cepstra plus Δ MFCC ($\alpha = 0.7$). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.88	0	0
2	0.3	4.57	0.6	4
3	1.5	4.7	2.4	39
4	0.5	4.46	1	1
5	1.3	4.85	1.6	6
6	0.4	5.17	0.8	1
7	0.4	4.43	0.8	8
8	1.7	4.1	3	17
9	0.2	5.16	0.4	1
10	0.2	4.63	0.4	2
11	4	2.9	35.2	12
12	0	4.79	0	0
13	3.2	3.79	8.4	31
14	0.3	4.96	0.6	9
15	1.2	5	2	4
16	4.3	2.83	10	47
17	2.1	4.87	2.6	53
18	0.6	5.54	1.2	4
19	6.5	2.38	37	65
20	6.2	2.07	41.2	28
21	0.6	4.49	1.2	9
mean	1.69	4.36	7.16	16.24

Table B.24: Cepstra plus Δ MFCC ($\alpha = 0.7$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	17.98	16	-4.47	-19.92
two	18.93	15.41	-3.07	-2.72
three	16.01	13.27	-1.14	-0.54
four	25.5	22.86	-6.84	-10.05
five	15.64	13.88	-1.26	-1.06
six	20.44	17.7	-3.84	-3.87
seven	15.49	12.89	-0.78	-0.08
eight	16.67	14.56	-1.84	-1.81
nine	13.18	9.95	-0.02	0.68
zero	13.83	10.76	-0.29	0.6
nought	16.2	14.48	-1.5	-1.42
oh	20.5	16.96	-3.95	-5.31
mean	17.53	14.89	-2.42	-3.79

Table B.25: Δ Cepstra plus MFCC (α = 0.8). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	17.98	16	-4.47	-19.92	56.35	85.95
2	11.91	9.68	-0.12	0.09	41.18	73.38
3	9.92	6.59	1.48	2.25	27.37	59.38
4	8.72	6.29	1.87	2.63	23.37	52.81
5	7.56	4.4	2.3	3.15	18.96	39.33
6	6.39	3.8	2.73	3.52	14.1	28.62
7	5.16	2.88	3.06	3.86	10.68	24.05
8	5.25	2.73	3.09	3.87	9.19	18.1
9	4.88	2.55	3.18	4.04	8.7	15.52
10	3.96	2.06	3.4	4.31	6.46	14.81
11	3.84	1.81	3.55	4.46	6.38	14.48
12	3.83	1.77	3.6	4.56	5.91	14.05

Table B.26: Δ Cepstra plus MFCC (α = 0.8). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.7	0	0
2	0.2	5.05	0.4	2
3	2.2	4.56	2.6	17
4	0.5	4.54	1	4
5	0.7	5.17	2.4	3
6	0	5.89	0	0
7	0	5.18	0	0
8	0.3	4.62	0.6	8
9	0.4	5.13	0.8	1
10	0	5.75	0	0
11	7.9	1.78	34.8	41
12	0	5.6	0	0
13	0.1	5.38	0.2	1
14	1	4.69	2.4	3
15	0.8	4.77	1.6	38
16	1.6	3.94	3.2	36
17	0.2	4.96	0.4	3
18	0.2	5.91	0.4	5
19	8.9	2.23	28.6	40
20	12.1	0.45	44.8	93
21	0	4.54	0	0
mean	1.77	4.56	5.91	14.05

Table B.27: Δ Cepstra plus MFCC ($\alpha = 0.8$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	22.97	21.13	-7.5	-16.56
two	25.16	21.51	-7.46	-8.74
three	21.14	17.73	-3.88	-3.81
four	34.16	32.22	-19.39	-25.22
five	21.65	20.8	-4.67	-7.06
six	29.73	26.32	-10.9	-15.08
seven	24.06	21.68	-5.33	-4.7
eight	20.54	18.31	-4.39	-18.32
nine	19.55	15.77	-2.64	-2.52
zero	21.32	17.28	-4.19	-2.78
nought	21.43	19.74	-5.09	-7.11
oh	29.45	25.78	-11.47	-37.03
mean	24.26	21.52	-7.24	-12.41

Table B.28: Δ Cepstra plus Δ MFCC (α = 0.7). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	22.97	21.13	-7.5	-16.56	72.78	91.62
2	18.4	14.78	-3.15	-3.15	55.29	88.43
3	14.97	10.89	-0.74	0.16	38.81	77.24
4	14.68	10.84	-0.65	-0.13	38.83	76.19
5	12.75	9.07	0.25	1.17	33.04	69.43
6	11.96	8.07	0.64	1.63	28.69	55.24
7	11.21	6.72	0.95	2.16	23.52	50.19
8	10.57	5.93	1.24	2.43	18.3	44.52
9	9.99	5.25	1.44	2.7	15.38	44.71
10	9.82	4.74	1.62	3.1	13.71	38.67
11	9.08	4.22	1.87	3.38	12.18	31.1
12	9.38	4.3	1.74	3.43	11.96	31.9

Table B.29: Δ Cepstra plus Δ MFCC (α = 0.7). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0.9	5.24	2.2	31
2	0.1	5.06	0.2	2
3	3.5	3.72	12.8	34
4	0	4.82	0	0
5	5.2	3.32	11	18
6	0	4.88	0	0
7	0.3	4.93	0.6	3
8	0.1	4.9	0.2	2
9	1.2	5.01	1.4	16
10	0	5.97	0	0
11	14.1	0.67	42.4	59
12	0	4.99	0	0
13	8.2	1.81	24.4	86
14	3.5	3.67	8.8	15
15	3.1	3.21	9.2	77
16	10.2	1.83	18.8	68
17	2.5	3.5	7.4	44
18	0.1	5.7	0.2	3
19	10.2	1.32	25.8	63
20	22	-5.07	70	97
21	5	2.63	15.8	52
mean	4.3	3.43	11.96	31.9

Table B.30: Δ Cepstra plus Δ MFCC ($\alpha = 0.7$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	20.29	17.74	-6.32	-51.43
two	14.82	12.24	-1.28	-0.87
three	16.13	13.58	-1.23	-0.68
four	23.5	19.79	-5.03	-4.66
five	14.11	11.66	-0.66	-0.15
six	17.03	13.84	-2.06	-3.26
seven	15.08	11.55	-0.75	0.35
eight	16.85	13.73	-2.1	-1.78
nine	13.24	10.69	-0.28	0.5
zero	14.45	12.15	-0.85	-0.29
nought	13.88	11.6	-0.73	-0.44
oh	14.4	12.88	-1.09	-5.39
mean	16.15	13.45	-1.86	-5.67

Table B.31: MFCC plus Δ MFCC (α = 0.5). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	20.29	17.74	-6.32	-51.43	62.97	91.81
2	11.41	9.81	0.06	-0.03	43.3	64.71
3	9.68	7.1	1.28	2.04	34.59	57.05
4	8.84	6.39	1.73	2.47	29.72	53.81
5	8.31	4.94	2.02	2.97	23.69	41
6	7.34	4.32	2.31	3.26	21.05	34.33
7	7	3.81	2.42	3.54	17.17	30.67
8	6.89	3.76	2.37	3.55	14.72	29.38
9	6.07	3.24	2.61	3.75	13.58	24.67
10	5.3	2.75	2.86	3.99	11.56	24.76
11	4.95	2.55	3.07	4.15	11.3	19.76
12	4.37	2.29	3.25	4.32	10.04	18.62

Table B.32: MFCC plus Δ MFCC (α = 0.5). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.9	0	0
2	1.1	4.83	1.2	25
3	1.8	4.49	3	29
4	5.4	2.49	30.8	44
5	0.2	5.21	0.4	3
6	0.1	5.5	0.2	1
7	2.5	3.85	4	21
8	3.6	3.16	7	43
9	0.5	5.1	1.2	1
10	0	5.42	0	0
11	6.9	2.43	26.8	18
12	0	5.51	0	0
13	1.1	5.11	3.8	4
14	1.9	4.72	6	7
15	0.6	4.83	1.2	3
16	2.1	4.75	5.6	35
17	0.2	5.03	0.4	3
18	2.2	4.65	3.6	21
19	7.2	1.8	53.8	58
20	10.7	0.94	61.8	75
21	0	5.01	0	0
mean	2.29	4.32	10.04	18.62

Table B.33: MFCC plus Δ MFCC ($\alpha = 0.5$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	14.8	12.95	-2.89	-11.74
two	10.5	9.79	0.65	0.75
three	12.55	10.69	0.17	0.64
four	21.71	17.67	-4.08	-3.57
five	11.5	9.49	0.48	0.7
six	12.93	10.37	-0.07	0.14
seven	9.21	7.28	1.43	2.04
eight	13.06	10.71	-0.42	0.11
nine	9.88	7.51	1.12	1.87
zero	9.49	8.25	1.37	1.87
nought	12.26	10.2	0.27	0.46
oh	11.2	9.35	0.32	0.06
mean	12.42	10.36	-0.14	-0.56

Table B.34: Cepstra plus Δ Cepstra plus MFCC. Equal weights. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	14.8	12.95	-2.89	-11.74	51.44	84.86
2	7.85	6.32	1.8	1.82	35	43.62
3	6.22	4.62	2.68	3.06	28.25	38.24
4	6.07	4.34	2.82	3.34	22.47	35.05
5	5.44	3.41	3.06	3.73	17.15	29.05
6	4.56	2.83	3.38	4	13.07	22.48
7	3.73	2.28	3.53	4.2	9.65	18.14
8	3.89	2.21	3.49	4.16	8.39	18.38
9	3.56	1.97	3.61	4.29	8.75	15.48
10	3	1.5	3.83	4.5	7.38	13.62
11	2.78	1.4	3.92	4.59	6.96	11.67
12	2.52	1.22	4.05	4.73	6.26	11.1

Table B.35: Cepstra plus Δ Cepstra plus MFCC. Equal weights. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.91	0	0
2	0.5	4.73	1	5
3	1.4	4.82	1.8	32
4	1.1	4	3.4	15
5	0.1	5.57	0.2	1
6	0	5.75	0	0
7	0.1	4.66	0.2	4
8	0.2	4.14	0.4	31
9	0.4	5.05	0.8	1
10	0	5.01	0	0
11	6.1	2.63	30.8	15
12	0	5.39	0	0
13	0.4	5.48	0.8	3
14	0.6	5.24	1.2	3
15	0.2	5.41	0.4	2
16	1.4	4.31	2.6	10
17	0.2	5.76	0.4	7
18	0.5	5.64	1	5
19	5.1	2.42	44.8	60
20	7.3	2.27	41.6	39
21	0	5.09	0	0
mean	1.22	4.73	6.26	11.1

Table B.36: Cepstra plus Δ Cepstra plus MFCC. Equal weights. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	14.8	12.95	-2.89	-11.74
two	10.5	9.79	0.65	0.75
three	12.55	10.69	0.17	0.64
four	21.71	17.67	-4.08	-3.57
five	11.5	9.49	0.48	0.7
six	12.93	10.37	-0.07	0.14
seven	9.21	7.28	1.43	2.04
eight	13.06	10.71	-0.42	0.11
nine	9.88	7.51	1.12	1.87
zero	9.49	8.25	1.37	1.87
nought	12.26	10.2	0.27	0.46
oh	11.2	9.35	0.32	0.06
mean	12.42	10.36	-0.14	-0.56

Table B.37: SS digit weights using Cepstra plus Δ Cepstra plus MFCC. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	14.8	12.95	-2.89	-11.74	51.44	84.86
2	7.77	6.17	1.97	2.05	33.66	43.19
3	5.88	4.35	2.85	3.24	24.96	37
4	5.48	3.73	3.07	3.6	17.88	32.95
5	4.65	2.87	3.31	3.96	15.31	24.67
6	3.97	2.43	3.65	4.22	11.59	20.48
7	3.11	1.86	3.77	4.39	8.1	17
8	3.24	1.87	3.72	4.34	7.45	16.43
9	3.07	1.7	3.78	4.44	7.89	14.19
10	2.51	1.36	4.01	4.65	6.66	12.1
11	2.46	1.13	4.1	4.74	6.13	10.33
12	2.29	1.05	4.21	4.87	5.78	10.05

Table B.38: SS digit weights using Cepstra plus Δ Cepstra plus MFCC. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.95	0	0
2	0.5	4.85	1	5
3	1.4	4.92	1.8	26
4	0.9	4.3	2.6	12
5	0.1	5.68	0.2	1
6	0	5.8	0	0
7	0.1	4.81	0.2	2
8	0.2	4.28	0.4	37
9	0.5	5.11	1.2	1
10	0	5	0	0
11	4	3.03	23.6	14
12	0	5.54	0	0
13	0.3	5.62	0.6	2
14	0.1	5.4	0.2	3
15	0.1	5.49	0.2	1
16	1.2	4.43	1.6	9
17	0.2	5.88	0.4	7
18	0.5	5.78	1	4
19	4.1	2.79	47.8	57
20	7.8	2.4	38.6	30
21	0	5.13	0	0
mean	1.05	4.87	5.78	10.05

Table B.39: SS digit weights using Cepstra plus Δ Cepstra plus MFCC. Results by Client.

Appendix C

Summary Tables for Chapter 5

Digit	SI EER	SS EER	SI TDM	SS TDM
one	13.66	11.83	-0.51	-0.23
two	12.78	11.73	0.28	0.47
three	11.63	10.09	0.8	1.06
four	17.95	17.01	-1.99	-2.69
five	12.28	10.83	0.31	0.42
six	13.47	12.26	-0.03	0.2
seven	11.74	9.53	0.66	1.05
eight	13.3	11.69	-0.4	-0.07
nine	9.49	8.22	1.42	1.82
zero	10.36	9.38	1.1	1.52
nought	12.92	11.77	-0.07	-0.03
oh	12.81	10.69	0.27	0.73
mean	12.7	11.25	0.15	0.35

Table C.1: DOP LPC Cepstra. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	13.66	11.83	-0.51	-0.23	45.61	84.43
2	7.34	6.13	2.26	2.72	22.81	61.52
3	4.73	3.75	3.18	3.6	10.78	43.38
4	4.16	3.03	3.45	3.94	9.87	39.48
5	3.94	2.77	3.54	4.11	8.3	33.76
6	3.11	1.97	3.77	4.28	6.94	25.76
7	3.15	1.73	3.78	4.34	5.42	22.29
8	2.84	1.44	3.8	4.33	4.76	22.29
9	2.54	1.26	3.92	4.46	4.02	18.81
10	2.28	0.97	4.08	4.63	2.98	14.9
11	2.33	0.95	4.1	4.66	2.93	12.62
12	2.12	0.79	4.19	4.77	2.37	12.33

Table C.2: DOP LPC Cepstra. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.75	0	0
2	0	4.33	0	0
3	0.3	5.54	0.6	20
4	0.7	4.67	1.4	21
5	1.3	4.78	2.4	13
6	0.4	5.51	0.8	2
7	0	4.62	0	0
8	0.7	4.18	1.6	45
9	0.4	4.95	0.8	4
10	0	4.71	0	0
11	1.5	3.7	12.6	22
12	0	5.27	0	0
13	0.2	5.54	0.4	3
14	0.7	4.64	1.4	15
15	0	5.36	0	0
16	3.7	2.81	11.4	50
17	0.1	5.97	0.2	2
18	0	6.44	0	0
19	4.3	3.16	12.6	50
20	2.2	3.73	3.4	11
21	0.1	4.47	0.2	1
mean	0.79	4.77	2.37	12.33

Table C.3: DOP LPC Cepstra. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	18.98	17.58	-3.18	-4.25
two	18.5	16.01	-2.24	-2.23
three	11.57	9.74	0.78	1.15
four	24.15	22.75	-5.65	-6.73
five	14.35	12.59	-0.46	-0.2
six	15.79	13.43	-1.2	-1.1
seven	15.69	14.56	-0.93	-0.77
eight	14.11	12.87	-0.42	-0.05
nine	13.26	11.41	0.06	0.31
zero	12.98	10.42	0.31	0.94
nought	13.06	12.43	0.08	0.11
oh	21.54	19.08	-4.5	-3.37
mean	16.16	14.41	-1.44	-1.35

Table C.4: DOP LPC ΔCepstra. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	18.98	17.58	-3.18	-4.25	60.04	92.57
2	11.98	10	0.51	1.1	40.66	74.76
3	6.74	4.69	2.6	3.3	20.31	47.14
4	6.15	4.11	2.79	3.68	15.69	41.81
5	4.99	2.94	3.11	4.08	9.36	30.86
6	3.55	1.91	3.52	4.44	8.7	20.62
7	3.03	1.36	3.8	4.69	6.46	15.33
8	2.88	1.23	3.73	4.62	4.9	13.14
9	2.89	1.12	3.77	4.69	4.7	9.24
10	2.23	0.69	4.01	4.92	2.76	8.19
11	1.96	0.57	4.21	5.15	1.92	6.33
12	1.88	0.5	4.25	5.25	1.72	7.1

Table C.5: DOP LPC ΔCepstra. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.48	0	0
2	0	5.27	0	0
3	0	6	0	0
4	0	4.91	0	0
5	0	5.02	0	0
6	0	6.58	0	0
7	0	5.41	0	0
8	0	4.53	0	0
9	0.4	4.89	0.8	10
10	0	6.32	0	0
11	2.8	3.8	19.6	27
12	0	5.74	0	0
13	0	6.03	0	0
14	0	5.35	0	0
15	0.1	5.72	0.2	1
16	0.2	4.82	0.4	21
17	0.6	4.96	2.2	2
18	0	7.04	0	0
19	1.3	4.54	5.6	31
20	5	2.77	7.2	49
21	0.1	4.14	0.2	8
mean	0.5	5.25	1.72	7.1

Table C.6: DOP LPC ΔCepstra. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	15.55	13.76	-1.96	-2.31
two	12.59	11.28	0.1	0.24
three	12.92	11.2	0.31	0.74
four	15.51	14.14	-1.11	-0.75
five	12.38	10.66	0.16	0.75
six	13.35	11.9	-0.27	0.15
seven	12.8	10.84	0.2	0.81
eight	14.86	13.21	-1	-0.56
nine	11.88	10.3	0.36	0.66
zero	11.13	10.28	0.57	0.92
nought	11.39	9.07	0.58	0.93
oh	11.8	10.65	0.16	0.34
mean	13.01	11.44	-0.16	0.16

Table C.7: DOP MFCC. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	15.55	13.76	-1.96	-2.31	53.87	91.81
2	8.32	6.57	1.65	2.12	31.67	56.62
3	6.41	4.47	2.54	3.15	19.87	42.71
4	5.05	3.26	3.05	3.71	14.5	42.33
5	4.69	2.7	3.25	4.02	11.78	32.81
6	4.12	2.34	3.49	4.21	11.26	28.33
7	3.74	2.05	3.56	4.39	9.36	24.24
8	3.8	1.99	3.49	4.34	8.45	24.67
9	3.73	2.01	3.58	4.44	7.86	21.95
10	3.39	1.64	3.76	4.61	6.94	19.62
11	3.29	1.58	3.9	4.73	7.13	17.14
12	2.81	1.22	4.06	4.87	5.66	14.9

Table C.8: DOP MFCC. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0.8	5.32	1.6	10
2	0.3	4.65	0.6	7
3	0.3	5.23	0.6	43
4	3.5	3.08	20.4	55
5	0	5.72	0	0
6	0.7	5.85	1.8	16
7	1.8	3.68	4.4	28
8	7	2.91	23.4	14
9	0.3	5.08	0.6	2
10	0	5.34	0	0
11	3.5	3.29	19.4	26
12	0	5.9	0	0
13	0	6.03	0	0
14	0.1	5.3	0.2	2
15	0	5.64	0	0
16	0.1	5.56	0.2	4
17	0	6.09	0	0
18	0	5.82	0	0
19	5	2.87	27.4	82
20	2.2	3.94	18.2	24
21	0	5.02	0	0
mean	1.22	4.87	5.66	14.9

Table C.9: DOP MFCC. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	22.38	19.56	-4.7	-4.96
two	18.28	15.88	-2.58	-1.85
three	17.65	13.81	-1.67	-0.6
four	22.31	20.61	-4.84	-4.71
five	16.18	14	-1.28	-0.9
six	20.17	17.18	-3.89	-2.66
seven	19.92	16.53	-2.96	-2.21
eight	20.54	15.73	-3.53	-1.63
nine	17.75	15.08	-2.2	-1.37
zero	16.7	13.2	-1.49	-0.55
nought	18.01	15.46	-1.99	-1.72
oh	20.03	16.37	-3.04	-2.53
mean	19.16	16.12	-2.85	-2.14

Table C.10: DOP ΔMFCC. Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	22.38	19.56	-4.7	-4.96	71.86	89.95
2	13.27	10.2	-0.04	0.93	49.16	74.29
3	9.79	6.42	1.48	2.76	24.76	57.57
4	8.9	5.05	1.92	3.22	19.66	46.57
5	7.3	3.76	2.39	3.79	13.17	41.24
6	7.01	3.08	2.6	4.12	9.91	34.19
7	6.26	2.42	2.89	4.41	8.36	26.57
8	6.57	2.09	2.79	4.5	6.99	24.43
9	6.06	1.78	2.95	4.65	5.46	22.48
10	5.52	1.44	3.21	4.91	3.8	23.43
11	4.62	1.2	3.54	5.09	3.31	17.43
12	4.57	0.98	3.67	5.29	2.46	15.67

Table C.11: DOP ΔMFCC. Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.38	0	0
2	0	5.74	0	0
3	1.5	5.12	2.8	15
4	0.7	5.13	1.4	33
5	0.3	4.68	0.6	40
6	0.2	6.22	0.4	6
7	1.4	4.6	4	35
8	3.1	3.59	3.6	55
9	0.6	5.4	1.2	8
10	0	6.3	0	0
11	0	5.64	0	0
12	0	6.69	0	0
13	0.3	5.36	0.6	4
14	1	5.72	4.2	3
15	0	5.19	0	0
16	1	5.04	5	2
17	0.6	5.54	1.2	8
18	0	6.58	0	0
19	1.7	4.72	4.6	12
20	4.9	3.68	17.4	55
21	3.3	3.65	4.6	53
mean	0.98	5.29	2.46	15.67

Table C.12: DOP ΔMFCC. Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	11.33	9.17	0.47	0.83
two	10.57	8.8	1.1	1.52
three	6.95	5.94	2.37	2.73
four	15.83	14.3	-1.03	-1.18
five	9.24	7.51	1.39	2
six	9.99	8.73	1.18	1.5
seven	8.39	7.19	1.74	2.09
eight	10.41	8.68	0.68	1.22
nine	7.53	5.94	2.21	2.64
zero	6.5	5.68	2.33	2.89
nought	8.98	7.78	1.73	2.01
oh	10.77	8.82	0.91	1.49
mean	9.71	8.21	1.26	1.64

Table C.13: DOP Cepstra plus DOP Δ Cepstra ($\alpha = 0.3$). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	11.33	9.17	0.47	0.83	38.49	77.43
2	6.06	4.24	2.93	3.54	19.47	50.24
3	2.78	1.74	3.96	4.5	6.28	24.9
4	2.66	1.45	4.14	4.78	4.08	17.81
5	2.16	1.16	4.18	4.88	2.79	11.71
6	1.64	0.82	4.41	5.04	3.16	5.81
7	1.57	0.49	4.46	5.11	3.07	4.52
8	1.51	0.53	4.38	4.99	2.45	4.52
9	1.41	0.43	4.4	5.06	2.53	3.33
10	1.02	0.3	4.56	5.22	1.65	2.52
11	1.14	0.25	4.65	5.33	1.33	2.62
12	1.11	0.21	4.71	5.41	0.97	2.24

Table C.14: DOP Cepstra plus DOP Δ Cepstra ($\alpha = 0.3$). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.36	0	0
2	0	5.02	0	0
3	0	6.14	0	0
4	0	5.34	0	0
5	0	5.31	0	0
6	0	6.29	0	0
7	0	5.28	0	0
8	0	4.65	0	0
9	0.1	5.43	0.2	1
10	0	5.63	0	0
11	1.1	4.12	12.4	7
12	0	5.72	0	0
13	0	6.21	0	0
14	0.1	5.39	0.2	1
15	0	6.13	0	0
16	0.7	4.28	1.6	2
17	0.1	6.32	0.2	2
18	0	7.14	0	0
19	1.8	4.16	4.8	28
20	0.5	3.92	1	6
21	0	4.72	0	0
mean	0.21	5.41	0.97	2.24

Table C.15: DOP Cepstra plus DOP Δ Cepstra ($\alpha = 0.3$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	11.53	9.58	0.24	0.53
two	10.08	8.72	1.28	1.52
three	9.87	8.58	1.39	1.7
four	13.97	12.59	-0.37	-0.05
five	9.79	8.66	1.08	1.46
six	11.21	9.69	0.72	1.05
seven	9.6	7.71	1.37	1.88
eight	11.76	10.4	0.09	0.54
nine	8.12	6.92	1.88	2.36
zero	8.53	7.15	1.93	2.35
nought	9.98	8.86	1.12	1.32
oh	9.94	8.71	1.07	1.52
mean	10.36	8.97	0.98	1.35

Table C.16: DOP Cepstra plus DOP MFCC ($\alpha = 0.7$). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	11.53	9.58	0.24	0.53	39.47	78.05
2	5.81	4.45	2.91	3.36	19.91	45.9
3	3.69	2.63	3.55	4	9.07	28.95
4	2.82	1.96	3.87	4.39	5.7	24.76
5	2.95	1.73	3.93	4.51	5.13	22.05
6	2.28	1.31	4.11	4.63	5.18	18.38
7	2.39	1.16	4.1	4.69	3.89	15.71
8	2.31	1.08	4.05	4.64	3.31	15.48
9	2.21	0.88	4.16	4.74	3.18	11.52
10	1.71	0.6	4.3	4.9	2.26	9.86
11	1.69	0.58	4.34	4.96	2.24	8.33
12	1.48	0.49	4.44	5.06	1.63	8.19

Table C.17: DOP Cepstra plus DOP MFCC ($\alpha = 0.7$). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	5.81	0	0
2	0	4.54	0	0
3	0.2	5.62	0.4	25
4	0.3	4.57	0.6	36
5	0.3	5.27	0.6	4
6	0.4	5.83	0.8	2
7	0	4.72	0	0
8	0.8	4.17	1.8	11
9	0.2	5.22	0.4	1
10	0	5.04	0	0
11	1	3.85	10.8	13
12	0	5.65	0	0
13	0	5.91	0	0
14	0.4	5.11	0.8	2
15	0	5.72	0	0
16	0.7	4.04	1.4	18
17	0	6.41	0	0
18	0	6.56	0	0
19	4.8	3.31	14	45
20	1.2	4.07	2.6	15
21	0	4.84	0	0
mean	0.49	5.06	1.63	8.19

Table C.18: DOP Cepstra plus DOP MFCC ($\alpha = 0.7$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	12.94	10.59	0.01	0.61
two	10.74	8.94	0.96	1.58
three	9.97	7.51	1.33	2.11
four	14.79	13.29	-0.57	-0.32
five	8.9	7.7	1.43	1.85
six	11.61	9.73	0.63	1.25
seven	10.17	8.07	1.24	1.9
eight	13	9.5	-0.14	1.06
nine	8.49	6.7	1.97	2.45
zero	7.99	6.22	1.99	2.66
nought	10.06	8.44	1.21	1.6
oh	9.78	7.5	1.19	1.99
mean	10.7	8.68	0.94	1.56

Table C.19: DOP Cepstra DOP plus Δ MFCC (α = 0.2). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	12.94	10.59	0.01	0.61	44.53	74.24
2	5.84	4.41	2.87	3.5	22.13	41.52
3	3.62	2.07	3.64	4.37	7.78	26.33
4	3.28	1.48	3.94	4.71	4.87	21.62
5	2.55	1.25	4.09	4.87	3.02	17
6	2.14	0.76	4.27	5.01	2.28	7.95
7	1.93	0.65	4.34	5.09	2.21	6.05
8	1.91	0.64	4.25	5.06	1.75	5.9
9	1.66	0.4	4.36	5.16	1.74	4.19
10	1.39	0.26	4.55	5.33	1.18	2.95
11	1.4	0.23	4.69	5.42	1.09	2.33
12	1.23	0.17	4.79	5.55	0.48	1.86

Table C.20: DOP Cepstra DOP plus Δ MFCC (α = 0.2). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.44	0	0
2	0	5.36	0	0
3	0	5.89	0	0
4	0.1	5.39	0.2	8
5	0.1	5.17	0.2	8
6	0	6.26	0	0
7	0	5.23	0	0
8	0	4.55	0	0
9	0	5.67	0	0
10	0	5.7	0	0
11	0.5	5.23	1	1
12	0	6.26	0	0
13	0	6.09	0	0
14	0.7	5.95	1.6	2
15	0	5.76	0	0
16	0.7	4.31	2	2
17	0.1	6.57	0.2	1
18	0	7.07	0	0
19	1.3	4.4	4.6	16
20	0.1	4.43	0.2	1
21	0	4.76	0	0
mean	0.17	5.55	0.48	1.86

Table C.21: DOP Cepstra DOP plus Δ MFCC ($\alpha = 0.2$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	14.64	12.56	-1.08	-1.67
two	13.83	11.38	-0.25	0.24
three	9.09	7.45	1.65	2.16
four	16.81	15.39	-1.46	-1.51
five	10.51	8.67	1.01	1.56
six	11.88	10.14	0.32	0.66
seven	11.09	9.73	0.92	1.3
eight	12	10.53	0.28	0.8
nine	9.92	8.03	1.41	1.79
zero	8.42	7	1.82	2.39
nought	9.42	8.35	1.46	1.76
oh	15.04	11.87	-0.71	0.3
mean	11.89	10.09	0.45	0.81

Table C.22: DOP Δ Cepstra plus DOP MFCC ($\alpha = 0.9$). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	14.64	12.56	-1.08	-1.67	48.08	82.62
2	7.78	5.96	1.97	2.62	29.66	61.24
3	4.39	2.85	3.37	4.05	11.69	28.1
4	3.64	2.16	3.65	4.46	7.21	22
5	3.13	1.5	3.83	4.72	5.32	16.57
6	2.19	0.97	4.13	4.95	5.5	7.76
7	1.68	0.68	4.33	5.15	4.1	6.05
8	1.79	0.67	4.18	5.01	3.56	5.24
9	1.95	0.55	4.21	5.05	3.84	3.9
10	1.41	0.36	4.4	5.24	2.07	2.71
11	1.16	0.3	4.58	5.43	1.7	2.52
12	1.15	0.25	4.66	5.54	1.61	2

Table C.23: DOP Δ Cepstra plus DOP MFCC ($\alpha = 0.9$). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.54	0	0
2	0	5.36	0	0
3	0	6.15	0	0
4	0	4.98	0	0
5	0	5.53	0	0
6	0	6.73	0	0
7	0	5.38	0	0
8	0.5	4.44	1.2	1
9	0.1	5.43	0.2	2
10	0	6.25	0	0
11	1.1	4.14	16.4	6
12	0	6.07	0	0
13	0	6.43	0	0
14	0	5.69	0	0
15	0	5.99	0	0
16	0	5.37	0	0
17	0.1	5.67	0.2	1
18	0	7.14	0	0
19	1.3	4.59	8.8	14
20	2.2	3.78	7	18
21	0	4.74	0	0
mean	0.25	5.54	1.61	2

Table C.24: DOP Δ Cepstra plus DOP MFCC ($\alpha = 0.9$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	16.23	13.82	-1.66	-1.64
two	14.62	11.64	-0.6	0.2
three	11.09	8.13	0.96	1.85
four	18.78	17.03	-2.65	-2.43
five	11.24	10	0.64	1.07
six	12.3	10.59	-0.01	0.44
seven	14.24	11.2	-0.16	0.5
eight	13.77	10.84	-0.36	0.7
nine	11.65	9.65	0.71	1.2
zero	10.28	7.66	1.07	1.96
nought	11.49	9.8	0.7	1.04
oh	16.62	12.52	-1.41	-0.16
mean	13.53	11.07	-0.23	0.39

Table C.25: DOP Δ Cepstra plus DOP Δ MFCC (α = 0.4). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	16.23	13.82	-1.66	-1.64	57.04	84.38
2	9.26	6.29	1.69	2.6	36.13	55.71
3	5.57	3	3.07	4.04	12.8	31.71
4	5.11	2.27	3.34	4.41	8.14	24.67
5	3.98	1.61	3.6	4.71	5.04	17.48
6	2.93	1.13	3.91	4.97	5.24	11.48
7	3.03	0.87	4.06	5.15	3.88	7.43
8	3.1	0.63	3.9	5.07	2.95	5.81
9	2.84	0.57	3.98	5.15	2.17	5.71
10	2.45	0.43	4.17	5.35	1.28	4
11	2.02	0.31	4.42	5.53	0.86	3.1
12	2.08	0.26	4.48	5.67	0.55	2.48

Table C.26: DOP Δ Cepstra plus DOP Δ MFCC (α = 0.4). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.73	0	0
2	0	5.78	0	0
3	0	6.02	0	0
4	0	5.38	0	0
5	0	5.18	0	0
6	0	6.76	0	0
7	0.1	5.34	0.2	1
8	0.1	4.54	0.2	1
9	0	5.8	0	0
10	0	6.58	0	0
11	0.5	5.21	1	3
12	0	6.47	0	0
13	0	5.94	0	0
14	0.1	6	0.2	1
15	0	5.81	0	0
16	0	5.33	0	0
17	0.1	5.8	0.2	2
18	0	7.21	0	0
19	0.8	5.15	2	8
20	3.4	3.73	7.2	19
21	0.3	4.31	0.6	17
mean	0.26	5.67	0.55	2.48

Table C.27: DOP Δ Cepstra plus DOP Δ MFCC ($\alpha = 0.4$). Results by Client.

Digit	SI EER	SS EER	SI TDM	SS TDM
one	16.23	13.78	-1.61	-1.4
two	13.66	10.95	-0.27	0.54
three	12.73	9.56	0.35	1.32
four	15.42	13.7	-0.96	-0.58
five	11.07	9.04	0.7	1.34
six	13.77	11.53	-0.61	0.31
seven	14.11	10.57	-0.06	0.93
eight	16.09	11.64	-1.33	0.32
nine	10.83	8.7	0.93	1.54
zero	10.8	7.7	1.05	2.04
nought	11.57	9.79	0.58	0.95
oh	13.47	10.61	-0.11	0.77
mean	13.31	10.63	-0.11	0.67

Table C.28: DOP MFCC plus DOP Δ MFCC ($\alpha = 0.1$). Single Digit Results Summary.

Sequence Length	SI EER	SS EER	SI TDM	SS TDM	SS ZFR	SS ZFA
1	16.23	13.78	-1.61	-1.4	58.5	82.19
2	8.61	6.3	1.86	2.68	33.15	59.14
3	6.05	3.39	2.9	3.85	13.75	38.9
4	5.31	2.6	3.31	4.25	9.29	29.1
5	4.28	1.86	3.55	4.56	7.2	25.43
6	3.78	1.49	3.72	4.79	4.79	18.33
7	3.45	1.09	3.87	5	3.94	12.57
8	3.66	0.97	3.72	4.98	3.69	10.14
9	3.24	0.84	3.86	5.1	3.08	8.33
10	2.72	0.57	4.07	5.31	1.96	8.29
11	2.26	0.57	4.33	5.46	1.57	5
12	2.14	0.38	4.46	5.62	1.11	4.57

Table C.29: DOP MFCC plus DOP Δ MFCC ($\alpha = 0.1$). Digit Sequence Results Summary.

Client	SS EER	SS TDM	SS ZFR	SS ZFA
1	0	6.45	0	0
2	0	5.71	0	0
3	0.5	5.71	1	5
4	0.3	4.95	0.6	23
5	0	5.39	0	0
6	0.1	6.57	0.2	1
7	0.8	4.61	1.6	12
8	2	3.91	5	22
9	0.1	5.66	0.2	1
10	0	6.18	0	0
11	0	5.39	0	0
12	0	6.76	0	0
13	0	6.13	0	0
14	0.5	6.11	1	1
15	0	5.73	0	0
16	0.2	5.6	0.4	1
17	0	6.35	0	0
18	0	6.76	0	0
19	1.5	4.59	7.6	6
20	1.5	4.57	5	16
21	0.4	4.79	0.8	8
mean	0.38	5.62	1.11	4.57

Table C.30: DOP MFCC plus DOP Δ MFCC ($\alpha = 0.1$). Results by Client.

Appendix D

Publications

1. † M.E. Forsyth and M.A. Jack.
   Discriminating semi-continuous HMM for speaker verification.
   In *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol I, pages 313-316, April 1994.

2. † M.E. Forsyth, P.C. Bagshaw, and M.A. Jack.
   Incorporating discriminating observation probabilities (DOP) into semi-continuous HMM for speaker verification.
   In *Proceedings ESCA workshop on Automatic Speaker Recognition, identification and Verification*, pages 19-22, April 1994.

3. M.E. Forsyth and M.A. Jack.
   Duration modelling and multiple codebooks in semi-continuous HMMs for speaker verification.
   In *Proc. European Conference on Speech Communication and Technology*, pages 319–322, September 1993.

4. M.E. Forsyth, A.M. Sutherland, J.A. Elliott, and M.A. Jack.
   HMM speaker verification with sparse training data on telephone quality speech.
   In *Speech Communication*, vol 13, pages 411-416, December 1993.

5. M.E. Forsyth, A.M. Sutherland, J.A. Elliott, and M.A. Jack.
   HMM speaker verification with sparse training data on telephone quality speech.
   In *Proceedings of the Fourth Australian International Conference on Speech Science and Technology (SST-92), Brisbane*, pages 67–72, December 1992.

†Reprinted in this appendix.

DISCRIMINATING SEMI-CONTINUOUS HMM FOR SPEAKER VERIFICATION.

M.E. Forsyth, M.A. Jack
Centre for Speech Technology Research
80 South Bridge, Edinburgh, EH1 1HN, SCOTLAND, UK

ABSTRACT

This paper describes the use of a multiple codebook SCHMM speaker verification system, which uses a novel technique for discriminative hidden Markov modelling known as discriminative observation probabilities (DOP). DOP can easily be added to a multiple codebook HMM system and require minimal additional computation and no additional training. The DOP technique can be applied to both speech and speaker recognition. Results are presented for text-dependent experiments on isolated digits from 27 true speakers and 84 casual imposters, recorded over the public telephone network in the United Kingdom.
DOP are shown to significantly improve speaker verification performance for several commonly used parameter sets.

1. INTRODUCTION

Semi-continuous hidden Markov Models (HMM) have previously been shown to be effective in the field of speech recognition [1], however this technique has only recently been applied to the field of speaker recognition [2, 3].
It has been shown that semi-continuous HMM (SCHMM) is superior to discrete HMM (DHMM) for speaker verification [3] and that state duration modelling (hidden semi-Markov models), and the use of multiple codebooks both provide significant benefits to a speaker recognition system [2].

This paper extends the work on multiple codebooks by testing a novel technique known as discriminating observation probabilities (DOP). The DOP technique is evaluated for cepstra, delta cepstra, mel-frequency cepstral coefficients (MFCC) and difference MFCC. DOP can be used in both speech and speaker recognition. Section 2 outlines the motivation and rationale for the DOP technique and section 3 describes the technique itself. Section 4 describes the database used in these experiments.

The multiple codebook SCHMM system and the novel technique used to isolate the speaker discriminating power of each codebook are described in section 5. The results are in section 6.

2. CONVENTIONAL MODELLING FOR SPEAKER VERIFICATION

The conventional way to apply HMM to the task of speaker verification is to make speaker-dependent models of a speaker. The verification procedure is then a matter of comparing the speech to be tested against the model. The Viterbi algorithm can be used to determine the probability of the speech having come from the model. If the probability is above a certain threshold the bid is accepted. The essence of this approach is *speech* modelling as opposed to *speaker* modelling. The probability of the speech coming from the model is, in a sense, a combined *speech* and *speaker* recognition probability. If the test data is noisy or distorted the false rejection rate will increase. This is because a noisy test utterance from a genuine speaker will not fit the speech model well, possibly leading to false rejection. Note that noise will not cause an imposter's speech to fit the speech model any better, and so will not increase the chance of false acceptance.

In some systems a normalisation technique has been used successfully to reduce the effect of speech modelling masking the speaker modelling . In particular it has been used to reduce the variation in speaker recognition scores caused by different telephone microphones [4]. This takes the form of an offset in the verification threshold which is proportional to the *speech modelling* quality of the test data. The size of the offset is determined by matching the test data with an independent set of reference models trained from speakers who are similar to the speaker whose identity is being verified.

Although normalisation has been shown to be a useful technique, it is simply a compensation for the fact that conventional HMM does not explicitly discriminate between speakers.

3. DISCRIMINATIVE MODELLING FOR SPEAKER VERIFICATION

In order to address the lack of explicit discrimination between classes in conventional HMM, a novel technique using discriminative observation probabilities (DOP) has been developed. The normalisation technique which is now commonly used in speaker verification is similar to, but significantly different from, a special case of DOP HMM. The procedure for generating a DOP HMM for a speaker (speaker A) is as follows.

- Train a conventional HMM for speaker A (model A)
- Train a conventional HMM as a reference model using appropriately chosen speech data (model R)
- Take the differences in the observation probabilities of model A and model R.
- Normalise the differences into probabilities in the range 0 to 1.
- Create a DOP model for speaker A by using these probabilities as the observation probabilities for the DOP model. The DOP model is not a separate model but is treated similarly to the various codebooks in a multiple codebook system

Proc. IEEE International Conference on Acoustics, Speech, Signal Processing, **I**, 313–316.

For these experiments the reference model was a general speaker independent model. The effect of this is that the new observation probabilities reflect what is different about speaker A compared to the rest of the population. If an acoustic observation occurred frequently in speaker A's training data but not so frequently in the speaker independent training data then the appearance of that acoustic observation in the test data is a good indication that the speech came from speaker A, and therefore the discriminating observation probability (DOP) is high. Likewise, if a codeword occurs frequently in the speaker independent training set but not in the training data of speaker A , then the appearance of that codeword in the test data is an indication that the speaker is not speaker A and so the DOP will be low. If the frequency of a codeword is similar for speaker A and for the speaker independent set then that codeword will not be useful in distinguishing speaker A and the DOP will be neutral (around 0.5).

DOP HMM has the following technical benefits

- A DOP model can be derived from a conventional HMM with no extra training
- The DOP model can be easily implemented as another information stream in a multiple codebook system.
- DOP models can be generated for all parameter sets in a multiple codebook system, doubling the number of information sources available for the verification decision.
- The information from the DOP model is at least partially independent from the information from the conventional model
- DOP models require minimal extra preprocessing.

3.1. Generalised DOP models

In these experiments the DOP models have been used to discriminate between a single speaker and a general speaker independent set. By choosing an appropriate reference model a DOP model can be created to maximise discrimination between any two groups of one or more speakers. For example, an obvious extension to this work would be to follow the approach used with normalisation and use a group of speakers who are are similar to speaker A to make the reference model, thereby maximising the discrimination between speaker A and speakers who sound like speaker A (cohort speakers). Note that if this would not be the same as normalisation because the segmentation is based on the true speaker model and not on the cohort model. Also DOP allows more flexibility in the codebook weighting than is possible with normalisation.

If the requirement of a system was to discriminate between male and female speakers, a model of male speakers could be trained and a model of female speakers used as the reference model.

The application of DOP models is not limited to discrimination between speakers. In speech recognition DOP models could be used to increase the distinction between commonly confused speech units. For example DOP models could increase discrimination between two phones or between a phone and a group of similar phones.

4. DATABASE

The data consists of twelve isolated digits (digits 'one' to 'nine' plus 'zero', 'nought' and 'oh'), recorded over the telephone, over a period of six months. A group 20 speakers (9 males, 11 females) are modelled by the system and an independent set of 84 imposter speakers is used for testing. There are 20 true speaker utterances and 84 impostor utterances in the test set for each digit. The data are all end-point detected to remove excess silence and minimise storage requirements.

The database is similar to the one used in [2] but with more speakers in the training set, and more occurrences of noisy or distorted data.

The codebooks used are of size 32 and were trained from an independent set of 20 speakers. The frame size was 20ms with 15ms overlap. The delta (first order difference) cepstra data used a window of 5 frames (current frame plus 2 frames either side).

4.1. Training

As in [2, 3], an emphasis has been placed in this work on using a minimal amount of training data, in the belief that the amount of training data will be strongly constrained in most large scale telephone applications, such as telephone banking.

Another significant factor is that the training data was recorded in a single session, whilst the test data was recorded over a period of six months. This is the most difficult case, since there can be significant variation in both the speakers voice and the telephone channel over different recording sessions. Five training tokens were used for each word model, with 6 states per model. A Gaussian distribution was used for duration modelling. The top six codeword probabilities for each speech vector were used in the HMM verifier.

The multiple codebook models were trained using only the cepstral codebook. All parameter sets were re-estimated but only the cepstral codebook was used to calculate the observation probabilities which were used to optimise the model in the Baum-Welsh algorithm. In other words, the cepstral codebook was used for segmenting the data into states in the baum-welsh re-estimation. This could lead to a advantage for the cepstral parameter set over the other parameter sets. For example, the performance of MFCC against the cepstral parameters may be different if the MFCC parameters were used for segmentation.

5. ISOLATING EACH PARAMETER

The verification process involves a Viterbi search through the silence/word/silence HMM lattice to determine the path with the highest probability. This *Viterbi path* is then used to calculate a verification score. The Viterbi path can be given as a frame interval defined by a beginning frame $t_{b,s}$, an end frame $t_{e,s}$ and a duration T_s for each state s of N states.

The system uses four parameter sets in four codebooks for verification (cepstra, delta cepstra, MFCC, delta MFCC). For training and for finding the Viterbi path during verification only the cepstra codebook is used.

It is not proposed that all these parameter sets would be used in a verification system. Part of the aim of this research is to determine which parameter sets have the best speaker discriminating ability. It is likely that some combination of some of the parameters will prove to be optimum

The DOP for each of the parameter sets are treated within the HMM as if they came from an another parameter set, although the DOP obviously use the same codebook as the parameter they are derived from. The cepstra DOP, for example, will use the same codebook as the normal cepstra observation probabilities.

The verification score is calculated as shown in equation 1. The duration probability $P(T_s/s)$ has a weighting d.

A codebook m of the C codebooks has a weighting c_m. The set of observations for the frame interval $t_{b,s}$ to $t_{e,s}$ for codebook m is denoted $O_{m,s}$.

$$\prod_{s=1}^{N}\left[P(T/s)^d \prod_{m=1}^{C} P(O_{m,s}/s)^{c_m}\right] \quad (1)$$

The probabilities from the front and back silence models are not included, as they contain no speaker discriminating information. For all experiments described here the duration weighting was kept fixed ($d = 0$).

Each parameter set in the multiple codebook system and its DOP counterpart was tested in isolation for verification performance. To do this the Viterbi path was found using the duration plus cepstra information which was used in training. The probability score that was calculated on the backtrace, was solely the contribution from the parameter being examined. The weightings for testing parameter i in isolation are shown in equation 5.

$$d = 0, \quad m \neq i \;\; c_m = 0, \quad m = i \;\; c_m = 1 \quad (2)$$

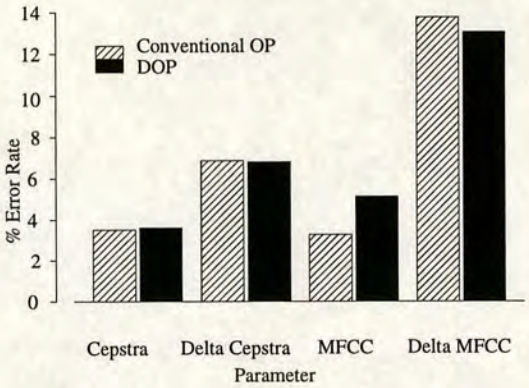

Figure 1. Comparison of each parameter set in isolation. Conventional HMM (striped) and the corresponding DOP on its own (black). 12 digit string EER for each parameter set

6. RESULTS

Figure 1 gives the EER for each parameter tested in isolation. The DOP all show significant speaker discriminating power, -comparable, in fact, to the conventional models. The test utterance consists of a concatenated sequence of the twelve isolated digits.

While the results in Figure 1 show that DOP have significant speaker discriminating power, the inclusion of DOP into a conventional HMM will only be useful if the speaker discriminating information of the DOP and the conventional observation probabilities are at least partially independent. In other words, if the conventional observation probabilities and the DOP make *different* errors then it may be possible to combine them to get a better result than is possible with either one alone.

Figure 2 show the difference in the EER between cepstra and DOP cepstra for each speaker. It can be seen from the

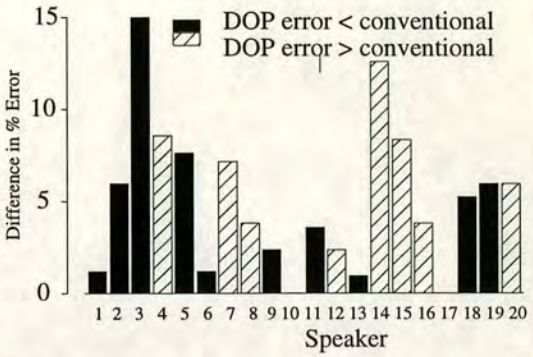

Figure 2. Independent speaker discriminating information. This plot shows the difference in conventional HMM and DOP HMM 12 digit sting EER for each speaker. Note that the two techniques have different strengths and weaknesses.

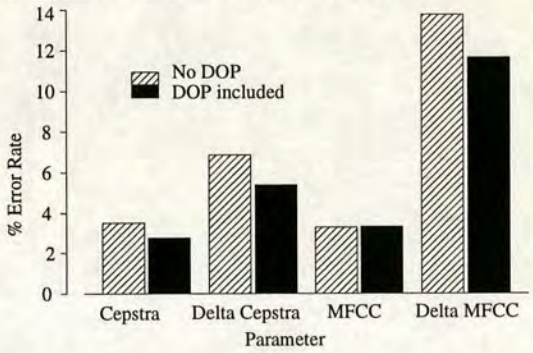

Figure 3. Comparison of conventional HMM (striped) and the conventional model with DOP included (black). 12 digit string EER for each parameter set

mix of light and dark bars that the two information streams *do* complement each other. For speakers {2, 3, 5, 18, 19} DOP offers significantly fewer errors, while for speakers {4, 7, 14, 15, 20} straight cepstra produces fewer errors. This is encouraging , since not only does DOP provide speaker discriminating information but it provides *new* information.

The next task is to combine the two information sources to produce a better EER. Initial attempts at including DOP into the conventional HMM using a weighted sum show that a clear advantage can be gained from the addition of DOP to the system for all the parameter sets. Figure 3 gives the comparative EER performance of the conventional model (striped) against the EER for the same parameter when the equivalent DOP are added (black).

Although equal error rates (EER) are the most common performance measure used in the literature, feedback from potential speaker verification users, such as banks, indicates that a negligible false rejection rate is crucial to the acceptability of a verification system [5] and so the zero false rejection (ZFR) error rate is perhaps a more useful measure of a systems performance. The ZFR rate is the false acceptance rate when the threshold is set such that there are no false rejection errors. The drawback of the ZFR rate is that it

Proc. IEEE International Conference on Acoustics, Speech, Signal Processing, **I**, 313–316.

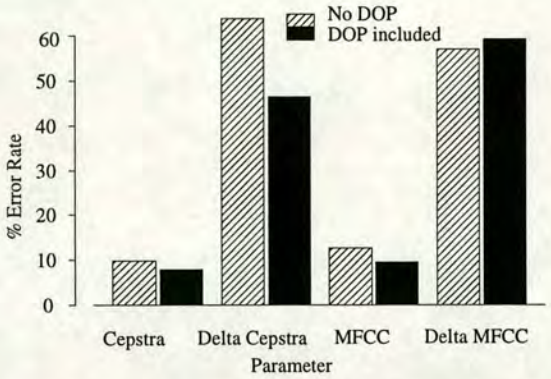

Figure 4. Comparison of conventional HMM (striped) and the conventional model with DOP included (black). 12 digit string ZFR for each parameter set

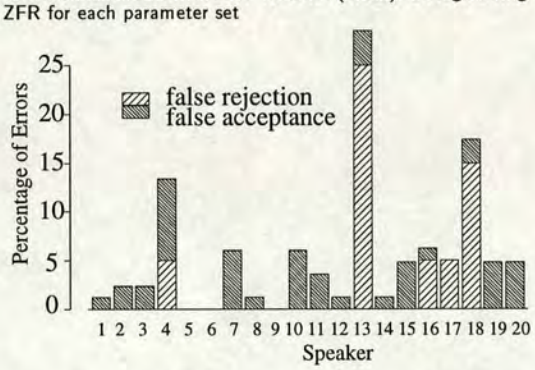

Figure 5. Breakdown of FR and FA errors by speaker, showing how the lack of speaker specific threshold increases the EER.

is sensitive to outliers in the database and so is not reliable when comparing systems using different databases. However if the database is the same it can be a useful measure for comparison.

Figure 4 illustrates the comparative ZFR rate performance of the conventional model (striped) against the ZFR rate for the same parameter when the equivalent DOP are added (black). There is clearly a general increase in performance as measured by ZFR when DOP are added. The weightings used for the results in Figure 3 were obtained from some simple trial and error experimentation, and are not optimal in any sense. They are , however , good enough to show that DOP is a useful addition to an HMM system. Optimal weightings could be obtained by many methods including discriminant analysis or by using a simple neural network. These approaches will be investigated in future work.

Some studies in the literature use speaker-specific thresholds to calculate EER results. Refer to citeForsyth93a for some discussion on why such EER are unlikely to be a realistic performance measure. In this work the EER thresholds are digit-specific but speaker independent. Figure 5 has a breakdown of false rejection (FR) and (FA) errors by speaker. The light bar represents FR errors and the dark bar represents FA errors. The potential advantage of using speaker-specific thresholds is clear. Thirteen out of the seventeen speakers with errors have fewer FR errors than FA errors. The other four speakers have far more FA errors than FR errors. This means that for each of the speakers with errors, the speaker independent threshold is either too low or too high. The difficulty in using speaker-specific thresholds arises from the limited amount of training data available. If a reliable threshold could be estimated for each speaker solely from closed test data a large improvement in performance could be gained.

7. CONCLUSIONS

DOP is a novel technique used to increase the discriminating power of HMM, which has been successfully used in a semi-continuous HMM speaker verification system to produce significant improvements in error rate. Although direct comparisons with other systems are not possible, due to the lack of a common database, the addition of DOP models shows a significant improvement over conventional HMM which are similar to those used in other systems [6, 7]. The technique is applicable to all applications of discrete, semi-continuous, or tied-mixture continuous HMM including speech recognition.

REFERENCES

[1] X. Huang, H. Hon, and K.-F. Lee, "Large-vocabulary speaker-independent continuous speech recognition with semi-continuous hidden Markov models," in *Proc. European Conference on Speech Communication and Technology*, vol. 1, pp. 163–166, Sept. 1989.

[2] M. Forsyth and M. Jack, "Duration modelling and multiple codebooks in semi-continuous HMMs for speaker verification," in *Proc. European Conference on Speech Communication and Technology*, 1993.

[3] M. Forsyth, A. Sutherland, J. Elliott, and M. Jack, "HMM speaker verification with sparse training data on telephone quality speech," in *Speech Communication*, Dec 1993. In press.

[4] A. Rosenberg, J. DeLong, C.-H. Lee, B.-H. Juang, and F. Soong, "The use of cohort normalised scores for speaker verification," in *International Conference on Speech and Language Processing*, 1992.

[5] "DTI biometrics workshop," report, U.K.Department of Trade and Industry, 24 June 1992.

[6] A. Rosenberg, C.-H. Lee, and F. Soong, "Sub-word unit talker verification using hidden Markov models," in *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, pp. 269–272, 1990.

[7] A. E. Rosenberg, C.-H. Lee, and S. Gokcen, "Connected word talker verification using whole word hidden Markov models," in *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, pp. 381–384, 1991.

Incorporating Discriminating Observation Probabilities (DOP) into Semi-Continuous HMM for Speaker Verification.

M.E. Forsyth, P.C. Bagshaw, M.A. Jack

Abstract—
This paper describes the use of a semi-continuous hidden Markov models for speaker verification. The system uses a technique for discriminative hidden Markov modelling known as discriminating observation probabilities (DOP). Results are presented for text-dependent experiments on isolated digits from 25 genuine speakers and 84 casual imposter speakers, recorded over the public telephone network in the United Kingdom. Performance measures which are used to assess the DOP technique are equal error rate, zero false rejection rate, zero false acceptance rate and two measures of the distance between probability distributions for genuine and imposter speakers. The different performance measures are assessed with regard to their suitability for comparing speaker verification algorithms. This analysis further supports previous work which shows that the addition of DOP to an HMM system provides a significant advantage in speaker verification performance.

Keywords— **Semi-continuous HMM, speaker verification, discriminating observation probabilities (DOP), telephone speech, text-dependent, isolated digits, performance measures**

1. Introduction

The technique of incorporating discriminating observation probabilities (DOP) into an HMM has been reported as being beneficial in the speaker verification task [3]. This paper extends that work by employing a second data set and several performance measures to test the reliability of the initial results.

Section 2 describes the database used in these experiments and describes how a second set of data is created by rotating the database to improve the robustness of performance measures. Section 3 briefly describes the DOP technique, which is introduced in [3].

The various performance measures used in this paper are explained in Section 4, including a new distance measure specifically aimed at assessing the performance of verification systems. The parameter sets used in these experiments are cepstra, mel frequency cepstra (MFCC), and the corresponding difference parameters. Each parameter set is tested separately and in combination with the corresponding DOP scores.

2. Database

The procedure for parameter extraction and for training the HMM is the same as described in [3]. The database is also the same, except for the addition of 3 speakers to the training set. There are now 23 speakers (12 female and 11 male) with the set of 84 imposters remaining the same.

The training database is divided into 5 blocks each containing 5 tokens per word. These blocks are labelled a to e. The A data set referred to in these experiments involves training on the a block and testing against the b,c,d,e blocks. The B data set involves training on the b block and testing on the a,c,d,e blocks. The C data set involves combining the results obtained from the A and B data sets.

There are 20 genuine speaker utterances and 84 imposter speaker utterances in the test set for each digit. The data was end-point detected to remove excess silence and minimise storage requirements.

The data consists of twelve isolated digits (digits 'one' to 'nine' plus 'zero', 'nought' and 'oh'), recorded over the U.K. telephone network. The training data was recorded in a single session, with the test data being recorded over a period of six months.

3. Discriminating Observation Probabilities (DOP)

In order to address the lack of explicit discrimination between classes in conventional HMM, a technique using discriminating observation probabilities has been developed [3].

The procedure for generating a DOP HMM for a speaker (speaker A) is as follows:

- Train a conventional HMM for speaker A (model A).
- Train a conventional HMM as a reference model using appropriately chosen speech data (model R).
- Take the differences in the observation probabilities of model A and model R.
- Normalise the differences into probabilities in the range 0 to 1.
- Create a DOP model for speaker A by using these probabilities as the observation probabilities for the HMM model.

The DOP model is not a separate model but is treated similarly to the various codebooks in a multiple codebook HMM.

For these experiments the reference model is a general speaker independent model, trained with data from an independent group of 20 speakers. A reference model is trained for each digit.

DOP HMM has the following technical benefits:

- A DOP model can be derived from a conventional HMM with no extra training

Mark Forsyth, Paul Bagshaw and Mervyn Jack are all with the Centre for Speech Technology Research, 80 South Bridge, Edinburgh, EH1 1HN, SCOTLAND, UK. E-mail: forsyth@cstr.ed.ac.uk pcb@cstr.ed.ac.uk maj@cstr.ed.ac.uk

- The DOP model can be easily implemented as another information stream in a multiple codebook system.
- DOP models can be generated for all parameter sets in a multiple codebook HMM, thus doubling the number of information sources available for the verification decision.
- DOP models require minimal extra processing.
- The results in Section 5 show that the combination of DOP scores and conventional HMM scores provides better speaker discriminating performance then either score alone.

4. Performance Measures

Speaker verification is concerned with the classification of unknown *bidders* into two classes, *genuine* speakers and *imposters*. There are two types of correct classification, the acceptance of genuine speakers, and the rejection of imposters. There are two corresponding types of errors, namely the rejection of genuine speakers, often called false rejection (FR), and the acceptance of imposters, often called false acceptance (FA).

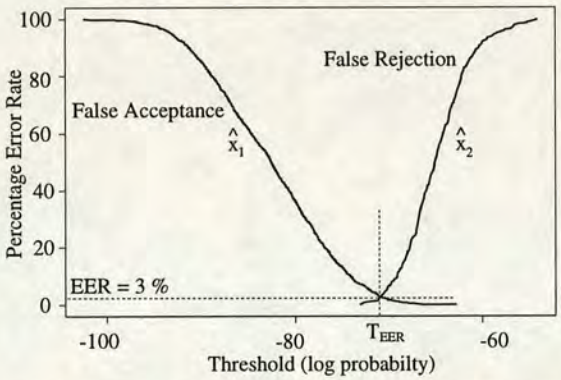

Fig. 1. Typical plot of FR rate and FA rate against choice of decision threshold. The EER, ZFR, and ZFA can be determined from this plot.

Figure 1 is a typical plot of FA rate and FR rate against the choice of decision threshold. Notice that there is a trade-off between FR and FA. Error rates for any given threshold can be determined from this plot. It is also possible for the trained eye to make some assessment of the robustness of the system to an imperfect choice of threshold. However, an objective measure of the separation of the genuine and imposter probabilities is still required to compare various algorithms and systems reliably.

There are several performance measures available for comparing speaker verification systems which measure different *aspects* of performance. The ZFR rate is the FA rate when no genuine speakers are rejected and the ZFA rate is the FR rate when no imposters are accepted. These measures are critically dependent on the worst genuine speaker score and the best imposter score, respectively. The ZFR and ZFA measures cannot be used as the sole basis for selecting one algorithm over another, since slight changes in the data could easily reverse the rankings of the algorithms, as can be seen in Section 5.2.

4.1 Equal Error Rate (EER)

The most common performance measure referred to in the literature is the equal error rate. This involves applying an *a posteriori* threshold T_{EER} which makes the percentage of FA and FR errors equal. It is important to make a distinction between whether T_{EER} is speaker-specific or speaker-independent [3], [4]. T_{EER} is speaker independent in these experiments.

The use of an EER implies a perfect choice of threshold, which is not possible in a real application since the threshold would have to be determined *a priori*. Therefore the EER provides an upper bound on performance and does not indicate how robust the system is to variations in data. Although EER is an important performance measure, it is also useful to have a measure of how well a system separates the probability distributions for the genuine speakers and the imposters. Such a measure would give an indication of the robustness of the system to an imperfect choice of threshold.

4.2 Mahalanobis Distance (MD)

A parametric measure of the distance between two statistical populations is the Mahalanobis distance [6], which assumes that the two populations have normal (Gauss-Laplacian) distributions. Consider that the two populations of log probabilities from imposter ($i = 1$) and genuine ($i = 2$) speakers are respectively represented by the sets,

$$x_i = \{x_{i,k} | k = 1, 2, \ldots, N_i\} \quad i = 1, 2 \tag{1}$$

These populations are normal distributions with a Lilliefors' probability [5] of approximately one, although it is noted that their greatest deviations from normal distributions are above the 90^{th}-percentile for the imposter scores ($i = 1$) and below the 10^{th}-percentile for the genuine speaker scores ($i = 2$).

The Mahalanobis distance of two univariate normal distributions is given by,

$$D^2 = \frac{(\bar{x}_1 - \bar{x}_2)^2}{\sigma_{12}} \tag{2}$$

where,

$$\bar{x}_i = \frac{1}{N_i} \sum_{k=1}^{N_i} x_{i,k} \quad i = 1, 2$$

$$\sigma_{12} = \frac{1}{N_1 + N_2 - 2} \sum_{i=1}^{2} \sum_{k=1}^{N_i} (x_{i,k} - \bar{x}_i)^2$$

The MD gives a measure of the separation between genuine speaker scores and imposter scores. Unfortunately, as is shown in section 5.1, this is not an ideal measure for the purpose of quantifying speaker discriminating performance. This is because the primary goal of a new algorithm is to reduce errors and most imposters are never mistaken for genuine speakers and most genuine speakers are not usually falsely rejected. Thus, the scores which most need to be improved are those near the equal error threshold.

The Mahalanobis distance assigns equal importance to all scores. A distance measure which targets the most important scores is required.

4.3 Targeted Distance Measure (TDM)

A figure of merit called the *targeted distance measure* is used in this paper. TDM targets the most important scores, namely the highest third of the imposter scores and the lowest third of the genuine speaker scores. It is calculated by the addition of two distance measures — TDM_{imp} for the imposter scores and TDM_{gen} for the genuine speaker scores.

$$TDM = TDM_{imp} + TDM_{gen} \tag{3}$$

where,

$$TDM_{imp} = 100. \left[\frac{1}{|\bar{x}_1 - \bar{x}_2|} . \frac{3}{N_1} \sum_{k=\lceil 2N_1/3 \rceil}^{N_1} (T_{EER} - \hat{x}_{1,k}) \right]$$

$$TDM_{gen} = 100. \left[\frac{1}{|\bar{x}_1 - \bar{x}_2|} . \frac{3}{N_2} \sum_{k=1}^{\lfloor N_2/3 \rfloor} (\hat{x}_{2,k} - T_{EER}) \right]$$

$$\hat{x}_{i,k} = k^{th}\text{member of } x_i \text{ sorted in ascending order}$$

This calculation takes an average *signed* distance from T_{EER} and normalises it with respect to the distance between the means of the two distributions. Note the reversal of sign between the calculation of TDM_{imp} and that of TDM_{gen}, so that a higher number corresponds to better performance in both cases.

5. Results

5.1 Comparing Performance Measures

Figure 2 is a comparison of 5 different parameter sets using seven different performance measures and three different data sets.

The ordinate measures performance, with the top representing the score of the best algorithm and the bottom representing the score of the worst algorithm. This means that the lowest error rates and the greatest distances are at the top. The ordinate is linear and has no absolute scale.

The seven performance measures, EER, ZFR, ZFA, TDM_{gen}, TDM_{imp}, TDM, and MD all have three vertical columns, one for each of the data sets. Each column has been normalised so that the *relative* performance of the five algorithms can be directly compared over all the performance measures, and all the data sets.

This figure is a comparison of performance measures as well as a comparison of algorithms. TDM shows a clear ranking of the algorithms. Not only is the ranking the same over all three data sets, but the relative differences in performance of the algorithms are the same over the three sets. This is an indication of a reliable performance measure, because it means that the relative merits of one algorithm over another can be assessed without undue sensitivity to the data set being used.

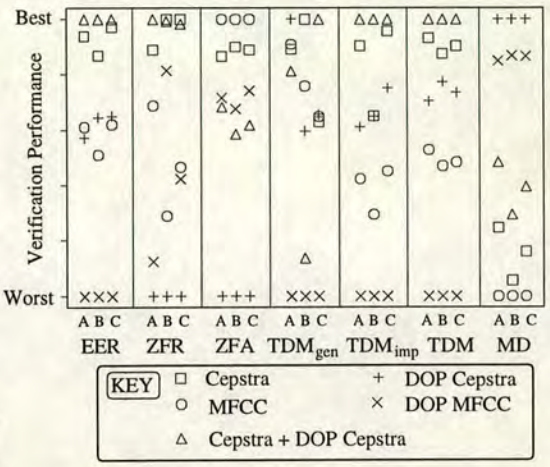

Fig. 2. Comparison of 5 different parameter sets using seven performance measures. Three different data sets are used, *A B* and *C*. The top of any vertical column represents the best algorithm for the given data set and performance measure.

Contrast this with the ZFR rate and the TDM_{gen} results. The relative positions of the algorithms change considerably between data set *A* and data set *B*, even though they are derived from the same database. These measures must therefore be used with caution.

The ZFA rate appears to be more reliable, although it should suffer from the same sensitivity as the ZFR rate, because it is a similar type of measure. It is interesting to note that the rankings from ZFA are different from the ranking of the other measures. This does not mean that it is a poor performance measure. It is a good measure of a different *aspect* of performance. The ZFA rate is a measure of system performance when security is the key requirement, taking priority over convenience and ease of use.

The Mahalanobis distance maintains the ranking for the different data sets and but does not appear to be measuring the same thing as the EER and the TDM. The MD favours the DOP algorithm in all cases. This means that the DOP scores are better separated overall than the conventional scores, but this has not lead to a corresponding reduction in real or potential misclassifications. This supports the need for the TDM.

Finally, the TDM_{imp} was more stable than the TDM_{gen}, which can probably be explained by the fact that TDM_{imp} is derived from 644 scores while TDM_{gen} is calculated from only 154 scores.

5.2 Adding DOP Scores to Conventional HMM scores

Several experiments were conducted using various combinations of normal cepstra and DOP cepstra. A simple weighted sum of the probabilities was employed, using the same method described in detail in [3]. Figure 3 shows the performance of the best of these combinations against

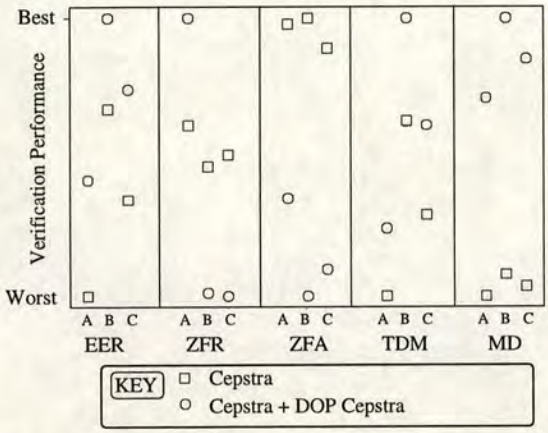

Fig. 3. Comparison of cepstra alone versus a weighted sum of cepstra plus DOP cepstra. Three different data sets are used, *A* *B* and *C* (which is the combination of the *A* and *B* sets). The top of any vertical column represents the best performance for the given performance measure.

cepstra alone. This figure differs from Figure 2 in that the results are normalised for each performance measure, instead of for each data set withing each performance measure. This allows some indication of the significance in the differences in the algorithms relative to the difference caused by using different data sets.

It can be seen that the EER, TDM and MD results were better for data set *B* than for data set *A*. As would be expected from a reliable performance measure, the results for data set *C* lie about half way between those for *A* and for *B*. The EER, TDM and MD performance measures all clearly illustrate the advantage of adding DOP to the system.

The results from ZFR and ZFA require some comment, since they illustrate the points made in Section 4. Cepstra without DOP gave clearly the best ZFA rate for all data sets, and on balance it was also superior for ZFR rate. It is interesting to note, however, that the *best* ZFR rate was obtained by DOP+CEP on data set *A* which other measures found to be the *hardest* of the data sets. This supports the proposition in Section 4 that these measures need to be used with caution.

The absolute values of the performance measures for the two algorithms can be seen in Table 5.2, along with the results for the other parameters tested. *No DOP* denotes only the conventional scores for that parameter were used, while *+DOP* denotes a combination of conventional and DOP scores. Note that since the TDM is a distance, the higher the number, the better the performance, while the reverse is true for EER.

The addition of DOP improves both performance measures for all the parameters tested. Comparison of these results with other studies in the literature [1], [7], [8] is not really possible because of the lack of a common database. Also note that state duration probabilities from the HMM have not been used in these experiments so that each algorithm can be examined in isolation. It has previously been shown that the inclusion of state duration probabilities significantly improves the EER [2].

TABLE I
THE EFFECTIVENESS OF INCLUDING DOP FOR SEVERAL DIFFERENT PARAMETERS. ALL VALUES ARE FOR THE *C* DATA SET.

Parameter	EER		TDM	
	No DOP	+DOP	No DOP	+DOP
Cepstra	2.95	2.49	3.53	3.69
Δ Cepstra	6.74	5.47	2.45	2.95
MFCC	3.88	3.86	3.42	3.49
Δ MFCC	12.71	11.19	0.36	0.97

6. CONCLUSIONS

A targeted distance measure has been developed which is a reliable complement to the conventional EER. It is easily calculated using the EER threshold. The TDM is a more useful measure for speaker verification than a total distance between the genuine and imposter probability distributions, such as the Mahalanobis distance.

The results of earlier work [3] on DOP HMM have been confirmed by experiments on a second data set. The incorporation of DOP scores lead to improvements in the EER and the TDM for a variety of parameters. Further investigation is required to find an optimal combination of multiple speaker discriminating information streams.

REFERENCES

[1] J. de Veth, G. Gallopyn, and H. Bourlard. Speaker verification over telephone channels based on concatenated phonemic hidden Markov models. In *Eurospeech*, volume 3, pages 2279–2282, September 1993.

[2] M.E. Forsyth and M.A. Jack. Duration modelling and multiple codebooks in semi-continuous HMMs for speaker verification. In *Proc. European Conference on Speech Communication and Technology*, pages 319–322, September 1993.

[3] M.E. Forsyth and M.A. Jack. Discriminating semi-continuous HMM for speaker verification. In *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, April 1994. (in press).

[4] M.E. Forsyth, A.M. Sutherland, J.A. Elliott, and M.A. Jack. HMM speaker verification with sparse training data on telephone quality speech. In *Speech Communication*. (in press).

[5] H.W. Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of American Statistical Association*, 64:399–402, 1967.

[6] P.C. Mahalanobis. On the generalized distance in statistics. *Proc. National Institute of Sciences of India*, 2(1):49–55, 1936.

[7] A.E. Rosenberg, C.-H. Lee, and S. Gokcen. Connected word talker verification using whole word hidden Markov models. In *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, pages 381–384, 1991.

[8] A.E. Rosenberg, C.-H. Lee, and F.K. Soong. Sub-word unit talker verification using hidden Markov models. In *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, pages 269–272, 1990.

References

Artieres, T., & Gallinari, P. (1993 Sept). Neural Models for Extracting Speaker Charateristics in speech Modelization Systems. *Pages 2263–2266 of: Proc. European Conference on Speech Communication and Technology.*

Assaleh, K. T., & Mammone, R. J. (1994). Robust Cepstral Feature For Speaker Identification. *Pages 129–132 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Atal, B. S. (1976). Automatic Recognition of Speakers From Their Voices. *Proc. IEEE*, **64**(4), 460–475.

Bach, R. (1970). *Jonathan Livingston Seagull.* Pan.

Bahl, L. R., Brown, P. F., de Souza, P. V., & Mercer, R. L. (1986). Maximum Mutual Information Information Estimation of Hidden Markov Model Parameters for Speech Recognition. *Pages 49–52 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Bahler, L. G., Porter, J. E., & Higgins, A. L. (1994). Improved Voice Identification Using A Nearest-Neighbour Distance Measure. *Pages 321–323 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A Maximisation Technique in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Ann. Math. Stat.*, **41**(1), 164–171.

Bennani, Y., & Gallinari, P. (1991). On the Use of TDNN-Extracted Features Information in Talker Identification. *Pages 385–388 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Bennani, Y., & Gallinari, P. (1994). Connectionist Approaches for Automatic Speaker Recognition. *Pages 95–102 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Bimbot, F., Chollet, G., & Paoloni, A. (1994 April). Assessment Methodology for Speaker Identification and Verification Systems. *Pages 75–82 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Bourlard, H., & Morgan, N. (1994). *Connectionist Speech Recognition: A Hybrid Approach.* Kluwer Academic Publishers.

Bridle, J. (1990). Alpha-net: a Recurrent Neural Network Architecture with a Hidden Markov Model. *Speech Communication.* Special Neurospeech Issue.

Buhrke, E. R., Cardin, R., Normandin, Y., Rahim, M., & Wilpon, J. (1994). Application of Vector Quantised Hidden Markov Modelling to Telephone Network Based Connected Digit Recognition. *Pages 105–108 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. I.

Campbell, J. P. (1995). Testing With the Yoho CD-ROM Voice Verification Corpus. *In: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.* Submitted for Publication.

Carey, M. J., & Parris, E. S. (1992). Speaker Verification Using Connected Words. *Pages 95–100 of: Proc. Institute of Acoustics*, vol. 14.

Chen, F., Millar, B., & Wagner, M. (1994). Hybrid Threshold Approach in Text-Independent Speaker Verification. *Pages 1855–1858 of: International Conference on Speech and Language Processing.*

Chou, W., Juang, H-H., & Lee, C-H. (1992 March). Segmental GPD Training of HMM Based Speech Recognizer. *Pages 473–476 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Chou, W., Juang, H-H., & Lee, C-H. (1993). Minimum Error Rate Training Based on the N-Best String Models. *Pages 652–655 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 2.

de Veth, J., Gallopyn, G, & Bourlard, H. (1993 Sept). Speaker Verification Over Telephone Channels Based on Concatenated Phonemic Hidden Markov Models. *Pages 2279–2282 of: Proc. European Conference on Speech Communication and Technology*, vol. 3.

Devillers, L., & Dugast, C. (1993 Sept). Combination of Training Criteria to Improve Continuous Speech Recognition. *Pages 2211–2214 of: Proc. European Conference on Speech Communication and Technology.*

Doddington, G. (1985). Speaker Recognition -Identifying People By Their Voices. *Proc. IEEE*, **73**, 1651–1664.

Eatock, J. (1992). *Speech Classification and Phoneme Performance in Speaker Recogntion.* Ph.D. thesis, University of Wales.

Eatock, J. P., & Mason, J. S. (1994). A Quantitative Assessment of the Relative Speaker Discriminating Properties of Phonemes. *Pages 133–136 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Epraim, Y., & Rabiner, L. R. (1988). On the Relations Between Modelling Approaches for Information Sources. *Pages 24–27 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Farrell, K., & Mammone, R. (1994). An Evaluation of Supervised and Unsupervised Clasifers for Speaker Recognition. *Pages 67–70 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Farrell, K., Mammone, R., & Assaleh, K. T. (1994). Speaker Recognition Using Neural Networks and Conventional Classifiers. *IEEE Trans. on Speech and Audio Processing*, **2**(1), 194–205.

Federico, A., & Paoloni, A. (1993 Sept). Bayesian Decision in the Speaker Recognition by Acoustic Parameterization of Voice Samples over Telephone Lines. *Pages 2307–2310 of: Proc. European Conference on Speech Communication and Technology.*

Floch, J-L Le, Montacie, C., & Caraty, M-J. (1994). Investigations on Speaker Characterisation From Orphee System Technics. *Pages 149–152 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Foil, J. T., & Johnson, D. H. (1983). Text-Independent Speaker Recognition. *IEEE Communications Magazine*, Dec., 22–25.

Forsyth, M.E., & Jack, M.A. (1993 Sept). Duration Modelling and Multiple Codebooks in Semi-Continuous HMMs for speaker verification. *Pages 319–322 of: Proc. European Conference on Speech Communication and Technology.*

Forsyth, M.E., & Jack, M.A. (1994 April). Discriminating Semi-continuous HMM for speaker verification. *Pages 313–316 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. I.

Forsyth, M.E., Bagshaw, P.C., & Jack, M.A. (1994 April). Incorporating Discriminating Observation Probabilities (DOP) into Semi-Continuous HMM for Speaker Verification. *Pages 19–22 of: Proceedings ESCA workshop on Automatic Speaker Recognition, identification and Verification.*

Furui, S. (1981). Cepstral Analysis Technique for Automatic Speaker Verification. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **29**(2), 254–272.

Furui, S. (1986). Research on Individuality Features in Speech Waves and Automatic Speaker Recognition Techniques. *Speech Communication*, **5**(2), 183–197.

Furui, S. (1994 April). An Overview of Speaker Recognition Technology. *Pages 1–9 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Gillick, L., & Cox, S. J. (1989). Some Statistical Issues in the Comparison of Speech Algorithms. *Pages 532–535 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Gish, H. (1990). Robust Discrimination in Automatic Speaker Identification. *Pages 289–292 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Gish, H., Schmidt, M., & Mielke, A. (1994). A Robust Segmental Method for Text Independent Speaker Identification. *Pages 145–148 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Godfrey, J. G., Holliman, E. C., & McDaniel, J. (1992 March). SWITCHBOARD: Telephone Speech Corpus for Research and Development. *Pages 517–520 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Godfrey, J. G., Graff, D., & Martin, A. (1994 April). Public Databases for Speaker Recognition and Verification. *Pages 39–42 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Hannah, M.J., Sapaluk, A.T., Damper, R.I., & Roger, I.M. (1993 Sept). The Effect of Utterance Length and Content on Speaker-Verifier Performance. *Pages 2299–2302 of: Proc. European Conference on Speech Communication and Technology.*

Hannah, M.J., Sapaluk, A.T., & Damper, R.I. (1994 April). The Effect of Utterance Length and Content on Speaker-Verifier Performance. *Pages 181–184 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Hattori, H. (1994). Text-Independent Speaker Verification Using Neural Networks. *Pages 103–106 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Hayakawa, S., & Itakura, F. (1994). Text-Dependent Speaker Recognition Using the Information in the Higher Frequencey Band. *Pages 137–140 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Hermansky, H., Morgan, N., Bayya, A., & Kohn, P. (1991). Compensation for the Effect of the Communication Channel in Auditory-Like Analysis of Speech (RASTA-PLP). *Pages 1367–1370 of: Proc. European Conference on Speech Communication and Technology.*

Higgins, A., Bahler, L., & Porter, J. (1991). Speaker Verification using Randomized Phrase Prompting. *Digital Signal Processing*, **1**(2), 89–106.

Higgins, A.L., Bahler, L.G., & Porter, J.E. (1993). Voice Identification Using Nearest-Neighbor Distance Measure. *Pages 375–378 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 2.

Hochberg, M. M., Renals, S. J., Robinson, A. J., & Cook, G. D. (1995 May). Recent Improvements to the ABBOT Large Vocabulary CSR System. *In: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.* Submitted for publication.

Huang, X.D., & Jack, M.A. (1988). Performance comparison between semi-continuous and discrete hidden Markov models of speech. *Electronics Letters*, **24**(3), 149–150.

Huang, X.D., Ariki, Y., & Jack, M.A. (1990). *Hidden Markov models for speech recognition.* Edinburgh University Press.

Irvine, D. A., & Owens, F.J. (1993 Sept). A Comparison of Speaker Recognition Techniques for Telephone Speech. *Pages 2275–2278 of: Proc. European Conference on Speech Communication and Technology.*

Itoh, K., & Saito, S. (1982). Effects of Acoustical Feature Parameters of speech on Perceptual Identification of Speakers. *Trans. IECE*, **J65-A**, 101–108.

Koehler, J., Morgan, N., Hermansky, H., Hirsch, H. G., & Tong, G. (1994). Intergrating RASTA-PLP into Speech Recognition. *Pages 421–424 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Lee, C-H., Rabiner, L. R., Pieraccini, R., & Wilpon, J. P. (1990). Acoustic Modelling for Large Vocabulary Speech Recognition. *Computer Speech and Language*, **4**, 127–165.

Levinson, S.E. (1986). Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition. *Pages 29–45 of: Computer Speech and Language.*

Lilliefors, H.W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of American Statistical Association*, **64**, 399–402.

Lipeika, A., & Lipeikiene, J. (1993 Sept). The Use of Pseudostationary Segments for Speaker Identification. *Pages 2303–2306 of: Proc. European Conference on Speech Communication and Technology.*

Liu, C-S, Lee, C-H, Juang, B-H, & Rosenberg, A. E. (1994 April). Speaker Recognition based on Minimum Error Discriminative Training. *Pages 325–328 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Mahalanobis, P.C. (1936). On the generalized distance in statistics. *Proc. National Institute of Sciences of India*, **2**(1), 49–55.

Matsui, T., & Furui, S. (1991). A Text-Independent Speaker Recognition Method Robust Against Utterance Variations. *Pages 377–380 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Matsui, T., & Furui, S. (1994a April). Similarity Normalization Method for Speaker Verification Based on A Posteriori Probability. *Pages 59–62 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Matsui, T., & Furui, S. (1994b). Speaker Adaptation of Tied-Mixture-Based Phoneme Models for Text-Prompted Speaker Recognition. *Pages 125–128 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Matsui, Tomoko, & Furui, Sadaoki. (1992a). Comparison of Text-Independent Speaker Recognition Methods using VQ-Distortion and Discrete/Continuous HMMs. *Pages 157–160 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Matsui, Tomoko, & Furui, Sadaoki. (1992b). Concatenated Phoneme Models for Text-Variable Speaker Recognition. *Pages 391–394 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. II.

McInnes, F. R. (1988). *Adaptation of Reference Patterns in Word-Based Speech Recognition.* Ph.D. thesis, University of Edinburgh.

McNemar, I. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psycometrika*, **12**, 153–157.

Millar, W., Oglesby, J., Pawlewski, M., & Tang, J. G. (1992). The assessment of Speaker Verification Systems. *Pages 423–430 of: Proc. Institute of Acoustics.*

Mokhtari, P., & Clermont, F. (1994). Contributions of selected Spectral Regions to Vowel Classification Accuracy. *Pages 1923–1926 of: International Conference on Speech and Language Processing.*

Naik, J. (1994 April). Speaker Verification over the Telephone Network: Databases, Algorithms and Performance Assessment. *Pages 31–39 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Naik, J. M., & Lubensky, D. M. (1994). A Hybrid HMM-MLP Speaker Verification Algorithm for Telephone Speech. *Pages 153–156 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Nolan, F. (1983). *The Phonetic Bases of Speaker Recognition.* Ph.D. thesis, Cambridge University.

Normandin, Y., Cardin, R., & Mori, R. De. (1994). High-Performacne Connected Digit Recognition Using Maximum Mutual Information Estimation. *IEEE Trans. on Speech and Audio Processing*, **2**(2), 299–311.

Oglesby, J. (1994 April). What's in a number?:Moving Beyond the Equal Error Rate. *Pages 87–90 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Oglesby, J., & Mason, J.S. (1990). Optimisation of Neural Models for Speaker Identification. *Pages 261–264 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Oglesby, J., & Mason, J.S. (1991). Radial Basis Function Networks for Speaker Recogntion. *Pages 393–396 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Openshaw, J.P., Sun, Z. P., & Mason, J. S. (1993). A Comparison of Composite Features Under Degraded Speech in Speaker Recognition. *Pages 371–374 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. II.

O'Shaughnessy, D. (1986). Speaker Recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Oct, 4–17.

Parris, E. S., & Carey, M. J. (1994). Discriminative Phonemes for Speaker Identification. *Pages 1843–1846 of: International Conference on Speech and Language Processing.*

Poritz, A. B. (1982 May). Linear Predictive Hidden Markov Models and the Speech Signal. *Pages 1291–1294 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Rabiner, L., & Hwang, B-H. (1992). *Fundamantals of Speech Recognition.* Prentice Hall International.

Rabiner, L. R., Wilpon, J. P., & Soong, F. K. (1989). High Performance Connected Digit Recognition Using Hidden Markov Models. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **37**(8), 1179–1213.

Rajasekaran, Y-H. Kao J. S. Baras P.K. (1993). Robustness Study of Free-Text Speaker Identification and Verification. *Pages 379–382 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. II.

Raman, V., & Naik, J. (1994). Noise Reduction for Speech Recognition and Speaker Verification in Mobile Telephony. *Pages 1839–1842 of: International Conference on Speech and Language Processing.*

Reynolds, D. A. (1992). *A Gaussian Mixture Modelling Approach to Text-Independent Speaker Identification.* Ph.D. thesis, Georgia Institute of Technology.

Reynolds, D. A. (1994 April). Speaker Identification and Verification using Gaussian Mixture Speaker Models. *Pages 27–30 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Rose, R. C., & Renolds, D. A. (1990). Text Independent Speaker Identification Using Automatic Acoustic Segmentation. *Pages 293–296 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Rosenberg, A. E. (1976). Automatic Speaker Verification: A Review. *Proc. IEEE*, **64**(4), 475–487.

Rosenberg, A. E., Lee, C.-H., & Gokcen, S. (1991). Connected Word Talker Verification Using Whole Word Hidden Markov Models. *Pages 381–384 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Rosenberg, A. E., Lee, C-H., & Soong, F. K. (1994). Cepstral Channel Normalisation Techniques for HMM-Based Speaker Verification. *Pages 1835–1838 of: International Conference on Speech and Language Processing.*

Rosenberg, A.E., & Soong, F.K. (1987). Evaluation of Vector Quantization Talker Recognition System in Text Independent and Text Dependent Modes. *Computer Speech and Language*, **2**(3/4).

Rosenberg, A.E., Lee, C.-H., Soong, F.K., & A.McGee. (1990a). Experiments in Automatic Talker Verification using Sub-Word Unit Hidden Markov Models. *Pages 141–144 of: International Conference on Speech and Language Processing.*

Rosenberg, A.E., Lee, C.-H., & Soong, F.K. (1990b). Sub-Word Unit Talker Verification using Hidden Markov Models. *Pages 269–272 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Rosenberg, A.E., DeLong, J., Lee, C.-H., Juang, B.-H., & Soong, F.K. (1992). The Use of Cohort Normalised Scores for Speaker Verification. *Pages 599–602 of: International Conference on Speech and Language Processing.*

Savic, M., & Gupta, S. K. (1990). Variable Parameter Speaker Verification System Based on Hidden Markov Modelling. *Pages 281–284 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing.*

Schiel, F. (1993 Sept). A Comparative Study of Speaker Adaptation under Realistic Conditions. *Pages 2271–2274 of: Proc. European Conference on Speech Communication and Technology.*

Soong, F. K., & Rosenberg, A. E. (1988). On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **36**(6), 871–879.

Sutherland, A., & Jack, M. (1988). *Aspects of Speech Technology*. Edinburgh University Press. Pages 197–200.

Thompson, J., & Mason, J. S. (1994 April). The Pre-detection of Error-prone Class Mambers at the Enrollment Stage of Speaker Recognition Systems. *Pages 127–130 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Tishby, N. Z. (1991). On the Application of Mixture AR Hidden Markov Models to Text-Independent Speaker Recognition. *IEEE Trans. on Signal Processing*, **39**(3), 563–570.

Tsoi, A. C., Shrimpton, D., Watson, B., & Black, A. (1994 April). Application of Artificial Neural Networks Techniques to Speaker Verification. *Pages 143–152 of: ESCA Workshop on Automatic Speaker Recognition Identification Verification.*

Wang, H-C., Chen, M-S, & Young, T. (1993). A Novel Approach to the Speaker Identification Over the Telephone Networks. *Pages 407–410 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. II.

Wilcox, L., Chen, F., Kimber, D., & Balasubramanian, V. (1994). Segmentation of Speech Using Speaker Identification. *Pages 161–164 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 1.

Yegnanarayana, B., Wagh, S. P., & Rajendran, S. (1994). A Speaker Verification System using Prosodic Features. *Pages 1867–1870 of: International Conference on Speech and Language Processing.*

Yu, G., & Gish, H. (1993). Identification of speakers engaged in dialog. *Pages 383–386 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. II.

Yuan, Z-X., Yu, C-Z, & Fang, Y. (1993). Text Independent Speaker Identification Using Fuzzy Mathematical Algorithm. *Pages 403–406 of: Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. II.